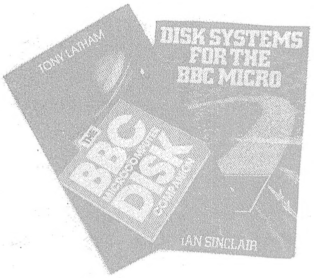
£1.00

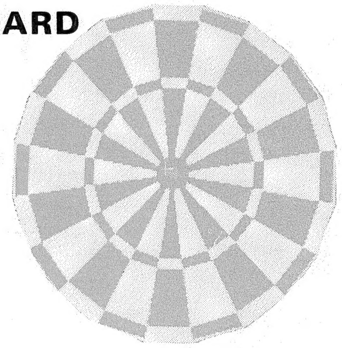
BEEBUG FOR THE BBC MICRO

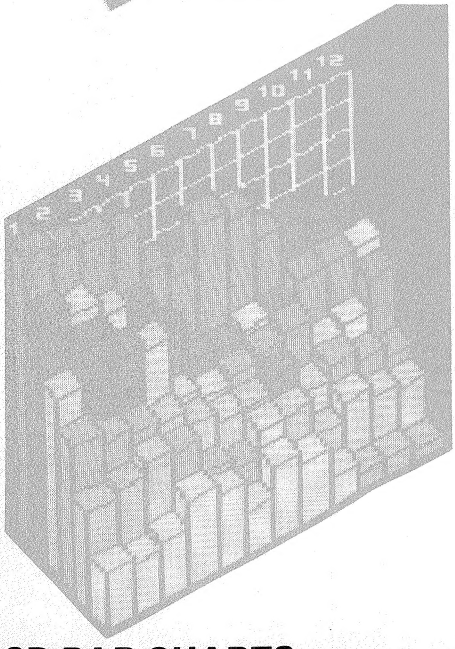
Vol 2 No 10 APRIL 1984


TONY LATHAM
DISK SYSTEMS FOR THE BBC MICRO
THE BBC MICROCOMPUTER DISK COMPANION
IAN SINCLAIR

Books for Disc Users Reviewed

DARTBOARD GAME





PLUS

- *** HOME ACCOUNTS – ANNUAL BUDGETS**
- *** ELEVASION GAME**
- *** INDEXING CASSETTE PROGRAMS**
- *** MOVING TEXT DISPLAY**
- *** APRIL FOOL**
- *** And much more**

3D BAR CHARTS

BRITAIN'S LARGEST COMPUTER USER GROUP
MEMBERSHIP EXCEEDS 25,000

EDITORIAL

THIS MONTH'S MAGAZINE

We are particularly pleased to include this month, a highly flexible and easy to use program for home accounting. This one allows you to analyse and plan your own home budget over a period of twelve months. It has been our intention for some time now to provide programs and articles that look at some of the more useful and helpful applications of the home micro, and this program is a significant step in that direction. We shall be looking at other applications for future issues and would welcome both your comments and suggestions on this.

This issue of the magazine appears very close to April Fools Day. We have managed to restrain ourselves and resist the temptation to play any jokes on you, the reader. But we have included a little routine that will enable you to have a some fun at the expense of your friends!

SECOND ANNIVERSARY ISSUE

Next month it will be two years since the founding of BEEBUG, and both we and the micro we support have come a long way in that time. To mark this second anniversary, we shall be giving away with every copy of the May issue (to members only) a £1 voucher which can be used in part payment for any BEEBUG or BEEBUGSOFT item (though not membership subscriptions or back issues). We shall also be including an additional machine code action game on the magazine cassette.

In the magazine itself, we shall be starting two new series for our third volume. One of these will be in the form of a Programmers' Workshop, which each month will present and explain a useful technique or procedure. The second will be a new series of articles aimed specially at the many hundreds of new members that are continuing to join BEEBUG. In both cases we would welcome any ideas, suggestions, requests and contributions. In this way we hope to provide an even better magazine that will cater alike for the new and the experienced user of the BBC micro.

We shall also be incuding with the next issue, a full index to the whole of Volume 2. This will be a separate insert so that you may readily include it in the binder with all your issues of Volume 2.

TICE BOARD NOTICE BOARD NOTICE BOARD NOTICE BOA

HINT WINNERS

This month's hint winners are W.G.R.Bain who wins the £1Ø prize, and R.Skemp who wins the £5 prize. We still need all your good hints and tips for the magazine.

MAGAZINE CASSETTE

This month, the magazine cassette contains the full versions of the Home Accounting (Annual Budgets) program and the Elevasion game. These include additional on-screen help and information. We have also included two 4-tone Mode 7 screen dumps for the Epson FX8Ø and MX8Ø printers. These are ideal for Teletext and Prestel screen displays and unlike some dumps, utilize the full paper width on 8Ø column printers.

BEEBUG MAGAZINE

GENERAL CONTENTS

Tested on Basic I & II and O.S. 1·2

COLOURFUL 3D BAR CHARTS(32K)

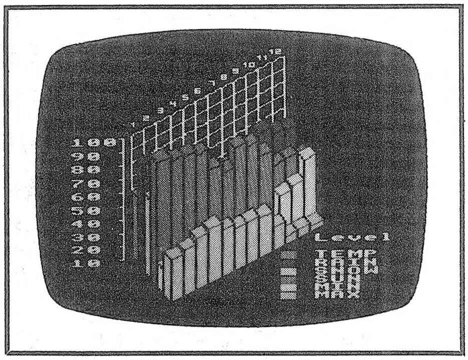
by J. Abboud

It is often much easier to understand data that is presented in a visual format rather than as columns of figures. The BBC micro, with its excellent graphics, is ideally suited to this task and here these capabilities are exploited to the full by the program 3DHIST to display 3D histograms or bar charts on the screen with three alternative styles of presentation.

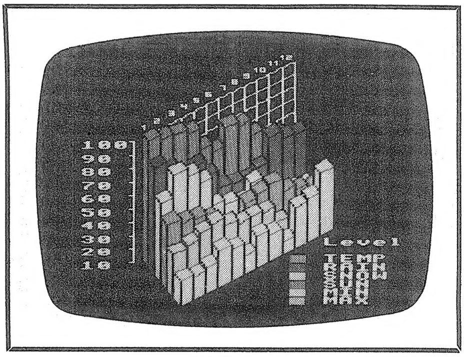
A bar chart or histogram is a simple way of displaying statistical data in the form of bars or columns. For example, a company might wish to display sales figures for each month throughout a year. This could be extended by breaking sales down into say six major types of product and then drawing separate bar charts or histograms for each. To make comparisons easier, we can then place all six together forming a 3D histogram. This is what the program here is designed to do.

The program itself could be used as a display routine in a larger program, or as it stands, possibly modified to allow keyboard entry of data. The version presented here is designed to illustrate three different ways of constructing such 3D histograms using data that is contained in DATA statements within the program. The program can be easily modified to suit your own particular needs. As written, it provides up to six levels of histogram each coloured differently, and with up to 12 data fields in each histogram. The program cycles through the three different forms of histogram, each plotted with a different ordering technique. In each case, pressing the space bar causes the program to cycle on to the next form of display.

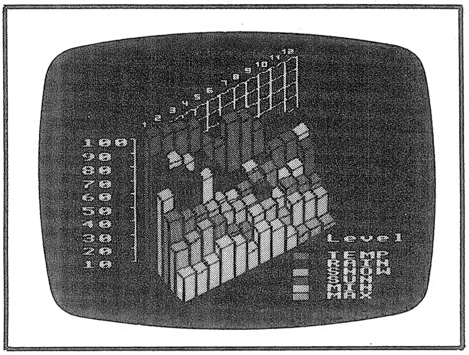
The first histogram displays the raw data without any kind of ordering. The resulting display may not be very clear as some of the background levels may be wholly or partially obscured by those in the foreground. The advantage is that the histograms are displayed in the order determined by the user (for example, histograms for a series of consecutive years in year order).



The second form of display orders the individual histograms by their average value or height. This will normally result in the higher histograms being placed at the back ranging down to the smallest at the front. This ordering is based on average values and thus individual fields may still be obscured by others nearer the front. This ordering may itself provide useful information. For example, monthly production figures for each of six years would be ordered so that the the year with the highest total production was at the rear



ranging down progressively to the year with the lowest annual production at the front.



In the third form of display, the columns for each individual field are now ordered from back to front. This tends to make it more difficult to compare whole histograms but does allow trends to be more strongly highlighted. For example, if production rose strongly from a low point at the start of a year to a very high level at the end of the year, this would be clearly indicated by this form of presentation. If required, the display will then cycle back to the first form of presentation again.

In all three cases, the histograms are displayed in a 3D perspective format, with vertical and horizontal scales and a colour key to identify each individual histogram. Just running the program and using the space bar to change the display will show you the excellent results that can be obtained. The three types of presentation are also illustrated in the screen shots that accompany this article.

PROGRAM NOTES

The program is well structured enabling you to make changes and modifications to suit your own needs. You can select the number of fields and levels by setting the values of level% and field% in the procedure PROCinit. Unless you use flashing colours or clever shading techniques, the number of levels is limited to six, the number of available colours. The number of fields is limited more by screen space and clarity but the arrays dimensioned at line 120 cater for a maximum of 12.

On the display, the 'X' axis is labelled with user defined characters (to get a small text size in mode 2). These are stored in the string F$, but could be changed for other characters such as the initial letters of the months of the year. The vertical scale of the graph is determined automatically by the values of highY% and lowY%, the highest and lowest points on the scale, which is divided into nine steps or increments. Both vertcal and horizontal scales are set up by PROCscale.

The data for the displays (six sets of twelve data items) is read into an array by PROCdata. The data is not scaled to the vertical scale shown on the screen display. Instead, a data value of 450 corresponds to the highest point on the vertical scale, and a value of 0 corresponds to the lowest. If you wanted to use other data and scale it for presentation then the following formula will do this for you:

```
Y=(450/(highY%-lowY%))*(X-lowY%)
```

where X is any given data item in the range from lowY% to highY% and Y is the calculated height for plotting in the program. Negative values will be plotted, though their appearance will be a little odd as the program was not designed to cope with such values.

In the program the type of display is selected by the value of the variable sort% (0, 1 or 2). If sort%=1 then the data is reordered using PROCsort, while if sort%=2 then the data is reordered by PROCsort1. One useful technique to note here is the use of the pointer array P%(6). All references to the data stored in the array B are made via this pointer array. Whenever the data is reordered, it is only the pointers that are shuffled around and never the data itself. This is much faster, as much less data has to be moved, and it also rather neatly enables the program to keep the right colour with the right field or level. This is a technique which can be very useful whenever the need arises to reorder large quantities of data and is standard computing practice.

»

PROCEDURES AND FUNCTIONS

1ØØØ	PROCinit	Initialise constants.
1Ø5Ø	PROCchars	Define characters.
13ØØ	PROCdata	Read data values.
156Ø	PROCgrid	Draw perspective grid.
168Ø	PROCscale	Display scales.
18ØØ	PROCdisplay1	Display histogram for third type of display.
185Ø	PROCdisplay	Display histogram for first two types of display.
19ØØ	PROChistogram	Draw histogram.
195Ø	PROCsort1	Order data by level.
2Ø1Ø	PROCsort	Order data by average.
211Ø	FNsum	Find sum of fields.
218Ø	PROCswap	Swap items in sort.

```
   1Ø REM Program 3DHIST
   2Ø REM Version B1.Ø
   3Ø REM Author J.Abboud
   4Ø REM BEEBUG April 1984
   5Ø REM Program subject to copyright
   6Ø :
  1ØØ MODE 2:VDU23,1,Ø;Ø;Ø;Ø;
  11Ø ON ERROR GOTO 35Ø
  12Ø DIM B(6,12),C%(6),L$(6),P%(6),sum
%(6)
  13Ø PROCinit:PROCchars:PROCdata
  14Ø REPEAT
  15Ø X%=25Ø:Y%=2ØØ:X1=Ø:Y1=45Ø:ANG=Ø.5
:D=5Ø*ANG:P=D:F%=25Ø:L%=2ØØ
  16Ø CLS:PROCgrid:PROCscale
  17Ø IF sort%=Ø THEN FOR I%=1 TO level
%:P%(I%)=I%:NEXT
  18Ø IF sort%=2 THEN 24Ø
  19Ø IF sort%=1 PROCsort
  2ØØ FOR J%=1 TO level%
  21Ø VDU 29,F%;L%;:PROCdisplay(C%(P%(J
%)))
  22Ø F%=F%+25:L%=L%-3Ø
  23Ø NEXT:GOTO 27Ø
  24Ø FOR J%=field% TO 1 STEP-1
  25Ø F%=25Ø:L%=2ØØ:PROCdisplay1(J%)
  26Ø NEXT
  27Ø VDU 4:PRINT TAB(15,23);"Level"
  28Ø FOR I%=1 TO level%
  29Ø COLOUR C%(I%):PRINT TAB(13,24+I%)
;CHR$237;:COLOUR 7:PRINT CHR$32;L$(I%)
  3ØØ NEXT:VDU23,1,Ø;Ø;Ø;Ø;
  31Ø sort%=(sort%+1)MOD3:Z%=GET
  32Ø UNTIL FALSE
  33Ø END
  34Ø :
  35Ø ON ERROR OFF:MODE7
  36Ø IF ERR<>17 THEN REPORT:PRINT" at
line ";ERL
  37Ø END
  38Ø :
 1ØØØ DEF PROCinit
 1Ø1Ø level%=6:field%=12:lowY%=1Ø:highY
%=1ØØ:sort%=Ø
 1Ø2Ø C%(1)=4:C%(2)=1:C%(3)=5:C%(4)=2:C
%(5)=6:C%(6)=3
 1Ø3Ø ENDPROC
 1Ø4Ø :
 1Ø5Ø DEF PROCchars
 1Ø6Ø LOCAL A$,I%,J%
 1Ø7Ø FORI%=224TO235
 1Ø8Ø VDU23,I%,Ø,Ø:READA$
 1Ø9Ø FORJ%=1TO11STEP2
 11ØØ VDU EVAL("&"+MID$(A$,J%,2))
 111Ø NEXT J%,I%
 112Ø VDU23,236,6Ø,66,153,145,145,153,6
6,6Ø
 113Ø VDU 23,237,Ø,255,255,255,255,255,
255,255
 114Ø B$=CHR$1Ø+STRING$(1Ø,CHR$8)
 115Ø F$="":FOR I%=1 TO field%:F$=F$+CH
R$(223+I%):NEXT I%
 116Ø ENDPROC
 117Ø DATA2Ø6Ø2Ø2Ø2ØØØ
 118Ø DATA7Ø1Ø7Ø4Ø7ØØØ
 119Ø DATA7Ø1Ø7Ø1Ø7ØØØ
 12ØØ DATA5Ø5Ø7Ø1Ø1ØØØ
 121Ø DATA7Ø4Ø7Ø1Ø7ØØØ
 122Ø DATA7Ø4Ø7Ø5Ø7ØØØ
 123Ø DATA7Ø1Ø1Ø1Ø1ØØØ
 124Ø DATA7Ø5Ø7Ø5Ø7ØØØ
 125Ø DATA7Ø5Ø7Ø1Ø7ØØØ
 126Ø DATA5CD454545CØØ
 127Ø DATA48D8484848ØØ
 128Ø DATA5CC45C5Ø5CØØ
 129Ø :
 13ØØ DEF PROCdata
 131Ø LOCAL A$,I%,L%,F%
 132Ø FOR I%=1 TO 6
 133Ø READ A$:L$(I%)=LEFT$(A$,5)
 134Ø NEXT
 135Ø FOR L%=1 TO level%
 136Ø FOR F%=1 TO field%
 137Ø READ B(L%,F%)
 138Ø NEXT F%,L%
 139Ø ENDPROC
 14ØØ DATA TEMP,RAIN,PRESS,SUN,MIN,MAX
 141Ø REM  (1)  (2)  (3)  (4)  (5)  (6)
 142Ø REM  ...  ...  ...  ...  ...  ...
 143Ø DATA 25Ø,23Ø,22Ø,19Ø,13Ø,11Ø
 144Ø DATA 18Ø,19Ø,2ØØ,15Ø,13Ø,11Ø
 145Ø DATA 4ØØ,36Ø,315,33Ø,27Ø,2ØØ
 146Ø DATA 19Ø,18Ø,2ØØ,29Ø,26Ø,25Ø
 147Ø DATA 3ØØ,38Ø,35Ø,3ØØ,19Ø,17Ø
 148Ø DATA 15Ø,2ØØ,13Ø,12Ø,1ØØ,Ø9Ø
 149Ø DATA 47Ø,45Ø,44Ø,42Ø,34Ø,3Ø5
 15ØØ DATA 37Ø,35Ø,29Ø,23Ø,23Ø,21Ø
 151Ø DATA 1ØØ,Ø9Ø,1ØØ,14Ø,12Ø,Ø7Ø
 152Ø DATA 12Ø,1ØØ,Ø5Ø,15Ø,16Ø,23Ø
 153Ø DATA 17Ø,18Ø,2ØØ,21Ø,18Ø,16Ø
 154Ø DATA 16Ø,15Ø,Ø9Ø,Ø3Ø,Ø3Ø,Ø2Ø
 155Ø :
```

»

```
 1560 DEF PROCgrid
 1570 FOR I%=1 TO 10
 1580 VDU 29,X%;Y%;:MOVE 0,0:GCOL0,7
 1590 FOR X=0 TO field%*50 STEP 20
 1600 Y=ANG*X:DRAW X,Y
 1610 NEXT:Y%=Y%+50
 1620 NEXT
 1630 VDU 29,X%;200;:MOVE 0,0
 1640 FOR I%=1 TO field%+1
 1650 DRAW X1,Y1:X1=X1+50:Y1=Y1+P:MOVE
X1,D:D=D+P
 1660 NEXT:ENDPROC
 1670 :
 1680 DEF PROCscale
 1690 step%=(highY%-lowY%)/9:y%=lowY%
 1700 GCOL0,7:VDU 5
 1710 MOVE -30,-10:DRAW -30,440
 1720 FOR I%=-10 TO 440 STEP 50
 1730 MOVE-45,I%:DRAW-30,I%:MOVE-240,I%
 1740 PRINT;y%:y%=y%+step%
 1750 NEXT
 1760 FOR I%=50 TO (field%)*50 STEP 50
 1770 Y%=ANG*(I%-50):MOVE I%-55,Y%+510:
PRINT MID$(F$,(I%)/50,1)
 1780 NEXT:ENDPROC
 1790 :
 1800 DEF PROCdisplay1(L):PROCsort1
 1810 FOR I%=1 TO level%
 1820 VDU 29,F%;L%;:PROChistogram((L-1)
*50,B(P%(I%),L),C%(P%(I%))):F%=F%+25:L%
=L%-30
 1830 NEXT:ENDPROC
 1840 :
 1850 DEF PROCdisplay(L)
 1860 FOR I%=field% TO 1 STEP-1
 1870 PROChistogram((I%-1)*50,B(P%(J%),
I%),L)
 1880 NEXT:ENDPROC
 1890 :
 1900 DEF PROChistogram(H,J,C)
 1910 MOVE 0,0:K%=ANG*H:GCOL0,C
 1920 MOVE H,K%:MOVE H,(K%+J):PLOT 85,(
H+50),(K%+J+25):MOVE (H+75),(K%+J-5):PL
OT 85,H,K%:MOVE (H+75),(K%-5):PLOT 85,(
H+75),(K%+J-5):MOVE (H+75),(K%-5):MOVE
(H+25),(K%-30):PLOT 85,H,K%:GCOL 0,0
 1930 DRAW H,K%+J:DRAW H+50,K%+J+25:DRA
W H+75,K%+J-5:DRAW H+25,K%+J-30:DRAW H,
K%+J:MOVE H+75,K%+J-5:DRAW H+75,K%-5:DR
AW H+25,K%-30:DRAW H+25,K%+J-30:MOVE H,
K%:DRAW H+25,K%-30:ENDPROC
 1940 :
 1950 DEF PROCsort1:LOCALII%,JJ%
 1960 FOR II%=1 TO (level%-1)
 1970 FOR JJ%=1 TO (level%-II%)
 1980 IF B(P%(JJ%),J%)<B(P%(JJ%+1),J%)
PROCswap(JJ%)
 1990 NEXT JJ%,II%:ENDPROC
 2000 :
 2010 DEF PROCsort:LOCAL II%,KK%
 2020 FOR KK%=1 TO level%
 2030 sum%(KK%)=FNsum(KK%)
 2040 NEXT KK%
 2050 FOR KK%=1 TO level%-1
 2060 FOR II%=1 TO level%-KK%
 2070 IF sum%(P%(II%))<sum%(P%(II%+1))
THEN PROCswap(II%)
 2080 NEXT II%,KK%
 2090 ENDPROC
 2100 :
 2110 DEF FNsum(U%):LOCAL sum
 2120 sum=0
 2130 FOR T%=1 TO field%
 2140 sum=sum+B(P%(U%),T%)
 2150 NEXT
 2160 =sum
 2170 :
 2180 DEF PROCswap(U%):LOCAL temp
 2190 temp=P%(U%):P%(U%)=P%(U%+1):P%(U%
+1)=temp
 2200 ENDPROC
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

INTERRUPT DRIVEN SPEECH - W.G.R.Bain

The following ammendments to the interrupt program at the end of the article on interrupt programming in BEEBUG Vol.2 No.6, will make the Acorn speech synthesiser pronounce figures and upper case letters when you type them on the keyboard. Lower case characters and other symbols will produce various words including some Basic Keywords:

```
  70 DATA &FF,&FF,0,0,0,0,0,0
1171 STA &72
1172 CMP #13
1173 BNE control
1174 .return
1180 LDA #114
1190 STA &72
1194 .control
1195 CMP #32
1196 BMI exit
1240 .exit
```

ACORN'S TELETEXT ADAPTOR

Reviewed by Mike Williams

This month we look in some detail at Acorn's Teletext Adaptor, which at long last is now available, and in fact being strongly advertised in many computer magazines. We also look at the BBC's plans to support the use of this unit and reveal some of the developments that are likely to take place in the future.

Product	:Acorn's Teletext Adaptor
Price	:£225 inc. VAT.
Supplier	:Vector Marketing, Dennington Estate, Wellingborough, Northants NN8 2RL.

The Teletext adaptor consists of a box half the size of the Beeb, which sits alongside it, and is connected to it by a 34-way ribbon cable to the micro's 1 MHz bus. There is also the Teletext Filing System (TFS) which comes as an EPROM to be fitted into a vacant sideways ROM socket. For anyone not sure of fitting the TFS EPROM, a voucher is supplied which can be taken (or sent) with the micro to any Acorn Service Centre, who will fit the TFS and also an O.S.1.2 ROM if necessary (and it is essential) free of charge. The unit is self powered and a comprehensive manual and Function Key label are also supplied.

SETTING UP

Once the Teletext File system is installed, and the ribbon cable plugged into the 1MHz bus, a TV aerial needs to be connected into the coaxial socket at the back of the Teletext adaptor, and the unit then switched on. Switching on the BBC micro gives 'Acorn TFS' as the current filling system. There is no visual indication of when the adaptor is switched on and its own power on/off switch is not very clearly marked. However, when you switch on the micro or select the Teletext Filing System (TFS) this checks the adaptor and displays the message 'Acorn TFS no power' if the unit is not powered up. One important point to note is that with TFS installed PAGE will automatically be set to &2400 on a disc system, or &1900 for cassette, even when the Teletext adaptor itself is not switched on. This means that you have nearly 3K less memory for your own programs, though this can be recovered when TFS is not being used.

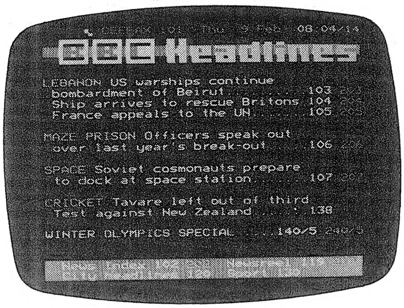
Once everything is connected and switched on, the first step is to tune the adaptor to the signals required. There are four separate channels marked 1, 2, 3, and 4 and it is desirable to tune these to BBC1 BBC2, ITV1 and ITV2 respectively. They can be tuned in a different arrangement but you are then likely to find later that, for example, selecting BBC1 from the keyboard will give ITV1 or something equally confusing.

TUNING

Typing *TELETEXT effectively turns your micro into a versatile Teletext receiver. All the features of this are controlled from the red function keys and also the cursor keys as we will see later. Pressing f4 allows a channel to be selected (Channel 1 to start with) and then pressing Shift/f4 gives a tuning scale for Channel 1. By slowly turning the red tuning wheel for Channel 1, I managed to find BBC1, after a little trouble, and the message 'Receiving Teletext' was displayed. At the same time a thermometer type signal strength scale appeared on the screen and the channel was fine tuned to give 100% on the scale. In practice the unit achieves 100% deflection very quickly but this does not guarantee that uncorrupted Teletext can be received, as the unit is very critical of the input signal. I found that I needed to fine tune my unit three times before I could get reliable reception. However, I did also manage to achieve reasonable reception with the simple loop aerial of the portable black and white TV I use quite often at home.

TELETEXT SERVICE

After tuning Channel 1, you proceed to tune the other three channels to BBC2, ITV1 and ITV2 respectively. You can then proceed to examine the Teletext pages on each service. First select the channel with f4 and then the

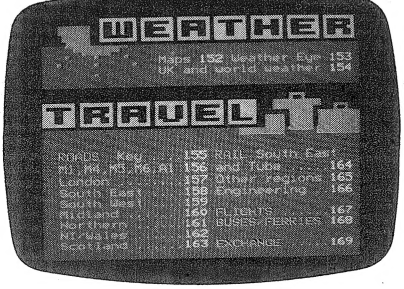


```
CEEFAX 101 Thu 9 Feb 08:04/14
BBC Headlines
LEBANON US warships continue
 bombardment of Beirut.........103
 Ship arrives to rescue Britons 104
 France appeals to the UN......105

MAZE PRISON Officers speak out
 over last year's break-out ...106

SPACE Soviet cosmonauts prepare
 to dock at space station......107

CRICKET Tavare left out of third
 Test against New Zealand .....138

WINTER OLYMPICS SPECIAL ....140/5
```

Ceefax News Headlines.

required page with f0 followed by the page number. As soon as that page is received you will see it displayed on the screen. The Teletext service provided by Ceefax and Oracle gives access to approximately 100 or more pages on each channel. These are numbered and broadcast in sequence so you normally have to wait until the page you have selected arrives.

The adaptor works in a very sophisticated manner and is an impressive performer with many features not found on normal Teletext receivers. The unit can 'Keep' pages in memory, up to 16 different ones at a time, and these are instantly available when called, instead of waiting for the cycle to come round again. In addition, these pages are constantly updated in memory. The unit can 'Save' pages to the Disc or Tape filing system so that they can be recalled at a later date. One point to note here is that a page is saved with its load address as given by its location in memory, not the screen address of &7C00. If you subsequently *LOAD a page you will see nothing displayed unless you specify the load address as &7C00 or change the saved load address to this value.

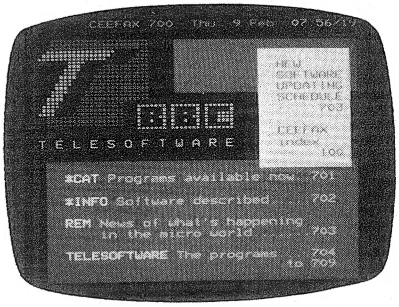


```
WEATHER
              Maps 152 Weather Eye 153
              UK and world weather 154
TRAVEL
ROADS Key .....155 RAIL South East
M1,M4,M5,M6,A1 156 and Tube       164
London         157 Other regions  165
South East     158 Engineering    166
South West     159
Midland        160 FLIGHTS        167
Northern       161 BUSES FERRIES  168
N.Wales        162
Scotland       163 EXCHANGE       169
```

Typical Ceefax Page.

PAGE LINKING

The Teletext adaptor will provide a full Teletext service with one major exception, and that is the ability to superimpose a Teletext page over the top of a normal TV picture. To compensate there are some extra facilities including the ability to handle 'linked' pages. This is still experimental but those of you with Teletext adaptors can try it out now. In principle any Ceefax page has the facility to be linked with other pages. For example, a page of headlines may have links to several other pages and each of these in turn may be linked to a set of more detailed news pages. Once



```
CEEFAX 700 Thu 9 Feb 07:56/17
                                NEW
                                SOFTWARE
                                UPDATING
                                SCHEDULE
                                     703
           BBC
                                CEEFAX
T E L E S O F T W A R E         index
                                     100
*CAT Programs available now     701
*INFO Software described        702
REM News of what's happening
     in the micro world         703
TELESOFTWARE The programs       704
                             to 709
```

Telesoftware Main Index.

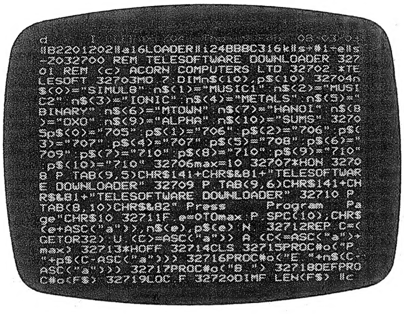
you have selected a Teletext page, any linked pages are loaded automatically and you can then follow the links by using the cursor keys to go up, down, left and right from the current page, giving much faster access than by conventional page selection. If you start with the Ceefax index on page 100 you can experiment with this new development right now.

DOWNLOADING PROGRAMS

The Teletext adaptor has a second mode of operation called Telesoft mode. In this mode, you can download programs from the BBC's Ceefax service to your micro. At present, up to 150K of software is available, changed fortnightly, though the overall capacity of Ceefax transmissions is scheduled to be increased by 50% from about April/May. These programs are

»

transmitted on pages 700 to 710 and are easily loaded using an Acorn downloader program found on page 704. You can also use this mode of operation to write



```
d      [illegible]                   OS [illegible]
‖B2201202‖a16LOADER‖i24888C316k‖s÷@1¬e‖s
-2032700 REM TELESOFTWARE DOWNLOADER 327
01 REM (c) ACORN COMPUTERS LTD 32702 *TE
LESOFT 32703MO.7:DIMn$(10),p$(10) 32704n
$(0)="SIMUL8":n$(1)="MUSIC1":n$(2)="MUSI
C2":n$(3)="IONIC":n$(4)="METALS":n$(5)="
BINARY":n$(6)="MTOWN":n$(7)="HANOI":n$(8
)="OXO":n$(9)="ALPHA":n$(10)="SUMS" 3270
5p$(0)="705":p$(1)="706":p$(2)="706":p$(
3)="707":p$(4)="707":p$(5)="708":p$(6)="
709":p$(7)="710":p$(8)="710":p$(9)="710"
:p$(10)="710" 32706max=10 32707*HON 3270
8 P.TAB(9,5)CHR$141+CHR$&81+"TELESOFTWAR
E DOWNLOADER" 32709 P.TAB(9,6)CHR$141+CH
R$&81+"TELESOFTWARE DOWNLOADER" 32710 P.
TAB(8,10)CHR$&82" Press    Program   Pa
ge"CHR$10 32711F.@=0TOmax:P.SPC(10);CHR$
(@+ASC("a")),n$(@),p$(@):N. 32712REP.C=(
GETOR32) U.(C>=ASC("a")) A.(C<=ASC("a")+
max) 32713*HOFF 32714CLS 32715PROC#o("P
"+p$(C-ASC("a"))) 32716PROC#o("E "+n$(C-
ASC("a"))) 32717PROC#o("B ") 32718DEFPRO
C#o(F$) 32719LOC.F 32720DIMF LEN(F$) ‖c
```

Typical Telesoftware Page.

your own programs to access the broadcast Teletext pages on all four channels. The programs transmitted so far have been mostly educational and are written to a high standard, although it has to be said that several of these programs in the past have not worked at the first transmission and have had to be modified during the fortnight that the programs were current.

IN THE PIPELINE

The BBC regard the Telesoftware service as part of an integrated system and will increasingly link the transmission of computer programs with the broadcasting of TV programmes. One example here is a new television series, presented by Ian McNaught-Davis and John Coll, called Computers in Control', which is all about robotics. The five programmes in the series will be shown on BBC1 at 12.30 on Fridays starting on 2nd March, with a repeat showing late on the following Thursday.

Other TV series will also be linked with the Telesoftware service and as a result some of the programs available will be changed every week instead of every fortnight as has been the case up to now. Future plans for Telesoftware also include the transmission of data files for interrogation by your micro as part of a widening information service. You'll be able to find out when all those interesting TV programmes on computers are being transmitted, even if you can't buy a copy of the Radio Times!

VALUE

Value for money? So far it is difficult to claim that it is good value for money. At £225 it is clearly overpriced, though it is technically advanced and there are no further costs involved in its use unlike Prestel. Tuning the unit is VERY sensitive as the red tuning wheels require only the most minute movement to go from ON tune to OFF tune and this can be very frustrating. The downloading of programs is exceptional and it is to be hoped that the high standard of programs broadcast, will be maintained.

If you have a TV and want access to Ceefax and Oracle then this is one very sophisticated way of achieving this with the added bonus of free Telesoftware. On the other hand, the home user is likely to find the Telesoftware service insufficient on its own to justify the £225 price tag. Despite these comments the BBC reckon that half the Teletext adaptors sold so far have gone to home users and the BBC's future developments in Telesoftware are something that I for one will certainly be watching with keen interest.

[I am most grateful for additional information and comments provided by Trevor Baker, and by Colin & Sue Cohen to whom many thanks.]

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

EASY ROM SELECTION - C.T.Marshall

It is possible to wire a zero insertion force socket to the keyboard, via ribbon cable from one of the internal ROM sockets, allowing easy replacement of ROMs from the vast amount of ROM software currently available.
[Note: Keep the ribbon cable short, otherwise extra buffering may be required for this socket.]

Tested on Basic I & II and O.S. 1.2

HOME ACCOUNTS - ANNUAL BUDGETS(32K)

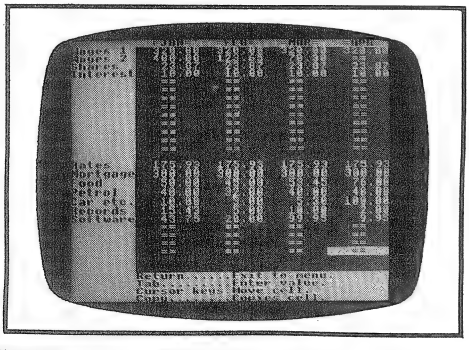
by David Reed

Many home computers are used for games playing but there are more useful applications for your micro. Helping to look after your money is one of these and although David Reed's program won't actually stop you going into the red it will help you to examine your monthly income and expenditure over a year and see where the money goes.

The program listed with this article really is designed to be easy to use. All the information that you need to supply is entered and modified directly on the screen so that you can see exactly what is happening. The monthly summaries of income and expenditure are quickly displayed and simple to follow. Working equally with disc or cassette systems, this is an extremely versatile and useful applications program.

You can use this program to enter all your major monthly incomings and outgoings over a period of a year, and examine your financial position month by month. You can start by entering your anticipated income and expenditure for each month, then save these to cassette or disc. As the months pass you can enter the actual amounts spent and received and update your forecasts. Although the program will handle sums to the nearest penny, its main purpose is to represent your present and future finances in round terms.

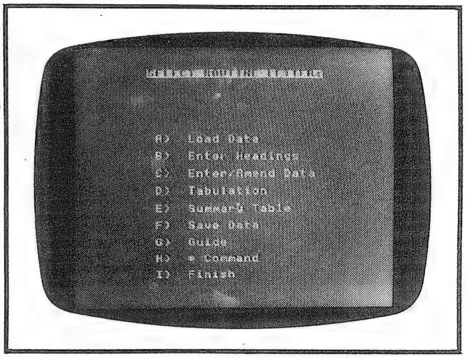
When you first run the program you will need to enter the current day's date so that your latest balance sheet can be dated. Any format will be accepted (or just Return for an undated entry) within the limit of 10 characters. You then move on to the program's main menu. The options available are described in the following paragraphs.



Tabulation of Credits & Debits.

CREDIT AND DEBIT HEADINGS

When you run the program for the first time all the credit and debit headings are so labelled. You can change and delete these as you wish. For example, you might wish to change the first credit heading to 'Salary', delete the remaining credit entries (by entering a blank string for each one) and then change some of the debit headings to 'Mortgage', 'Rates', 'Food', 'Petrol', etc, as you wish. Use the up and down cursor keys to select the heading, and Tab and Return to start and finish any text entered. These headings will be saved with your data and can always be returned to and changed at any time.



Main Program Menu.

DATA ENTRY

When you select the screen for data entry, your credit and debit headings will be displayed down the left hand side of the screen with the months across the top. The four cursor keys can be used to select the month and a credit or debit entry. The screen shows entries for four months at a time and these are changed by just moving the cursor keys left or right. Again Tab

»

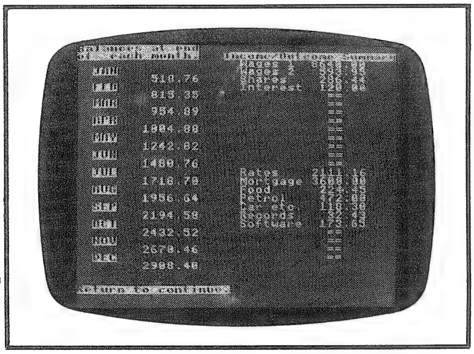
and Return are used to mark the start and finish of any amount entered. You can also use the Copy key to copy any amount immediately to the left of the current position (apart from January) and the Delete key will remove the amount in the current position, giving great flexibility.

TABULATION OF CREDITS AND DEBITS

At any stage you can tabulate all credits and debits on a month-by-month basis with the end of the month balance carried forward to the next month. Again four months are shown at a time, cycling through each following group of four months by pressing the Space Bar. This is the most detailed presentation of your finances.

SUMMARY TABLE

This summarises your annual budget by displaying end of the month balances throughout the year and the total amount for each of your credit and debit headings.



```
Balances at end        [illegible]
of each month.
                         Wages      [illegible]
JAN                      Wages 2    [illegible]
        518.76           Shares     [illegible]
FEB                      Interest   120.00
        815.35
MAR
        954.09
APR
       1004.80
MAY
       1242.62
JUN
       1480.76
JUL                      Rates      [illegible]
       1718.70           Mortgage   3600.00
AUG                      Food       [illegible]
       1956.64           Petrol     [illegible]
SEP                      Car etc.   [illegible]
       2194.58           Records    [illegible]
OCT                      Software   173.63
       2432.52
NOV
       2670.46
DEC
       2908.40

Return to continue.
```

Summary Table.

SAVING AND LOADING DATA

Once entered, all your data including your own credit and debit headings can be saved in a file on cassette or disc. You simply enter the filename when prompted. Data once saved can always be loaded back into your micro at a later date for further modifications and display.

* COMMANDS

This menu option allows any usual * command to be entered and acted upon without leaving the program, a very useful feature, particularly for disc users.

GUIDE

The version of the program on the magazine cassette also contains an on-screen guide to all the menu options for quick referance. The guide is omitted from the printed version to keep the program to a reasonable length.

PROGRAM NOTES

There are two particular areas of the program that you can tailor to your own needs. These are lines 2100 and 2120, both in the initialisation procedure PROCinit. Line 2100 sets up F% and B%, and these are used as the foreground and background colours respectively when the program is running. If the colour combination already given (i.e. yellow on blue) is not to your liking, then all you have to do is to alter these two values. The second area for adjustment, at line 2120, is for you to alter the number of debit and credit entries. This is done by specifying two things: the total number of entries (i.e. the number of debit plus the number of credit entries) and the point at which the split from credit to debit is to be made. SPLIT% gives the number of the first debit entry, and NUM% (not the array NUM%, just the single variable) gives the total number of entries.

Note also that when saving and loading to and from tape, the program provides only limited prompts. This is to keep the size of the program to a minimum.

```
   10 REM Program DOMESTIC ACCOUNTS
   20 REM Authors David Reed/David Fell
   30 REM VERSION B1.9M
   40 REM BEEBUG APRIL 1984
   50 REM PROGRAM SUBJECT TO COPYRIGHT.

   60 :
  100 ON ERROR PROCerror1:END
  110 MODE4:PROCinit:PROCtitle
  120 ON ERROR PROCerror2:IF E% END
  130 MODE4:PROCcolours:PROCcolour(1)
  140 PROCcntr("SELECT ROUTINE LETTER:"
,3)
  150 PROCcolour(2):RESTORE 4070
  160 FORI%=1TO8:READ A$
  170 PRINTTAB(10,2*I%+9);CHR$(64+I%);"
)  ";A$
  180 NEXT:PROCcolour(2)
  190 R=GET AND &DF
  200 IF R=65 CLEAR:R=0:DIM BCF%(13):pa
d$="   ==    ":PROCload:PROCmonths
```

»

```
 210 IF R=66 PROCtitles
 220 IF R=67 PROCentervalues
 230 IF R=68 PROCtab
 240 IF R=69 PROCsummary
 250 IF R=70 PROCsave
 260 IF R=71 PROCos
 270 IF R=72 PROCquit
 280 *FX4
 290 GOTO 130
1000 DEF PROCtitle
1010 PROCcolour(1)
1020 PROCspread("Domestic Accounts.",6)
1030 PROCcolour(2)
1040 PROCcntr("Annual Budgets.",11)
1050 PROCdate
1060 PROCcntr("Press SPACE to continue
.",23)
1070 REPEAT UNTIL GET=32:ENDPROC
1080 :
1090 DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,
AUG,SEP,OCT,NOV,DEC
1100 :
1110 DEF PROCcolour(X%)
1120 IF X%=2 COLOUR1:COLOUR128
1130 IF X%=1 COLOUR0:COLOUR129
1140 ENDPROC
1150 :
1160 DEF PROCcntr(A$,P%)
1170 PRINTTAB((40-LENA$)DIV2,P%)A$
1180 ENDPROC
1190 :
1200 DEF PROCspread(A$,P%)
1210 LOCALI%,B$:FORI%=1TOLENA$
1220 B$=B$+MID$(A$,I%,1)+" ":NEXT
1230 PROCcntr(B$,P%):ENDPROC
1240 :
1250 DEF PROCentervalues
1260 PROCcolour(1)
1270 VDU28,0,31,7,0,12,10
1280 LOCAL J%,K%,X%,Y%:FORI%=1 TONUM%
1290 IF I%<>SPLIT% PRINTHDG$(I%); ELSE
PRINT'HDG$(I%);
1300 NEXT:VDU28,8,31,39,0
1310 X%=1:Y%=1:P%=1:LP%=P%
1320 PROCdisplayvalue:*FX4,1
1330 REPEAT K%=GET
1340 PROCdisplaysinglevalue
1350 IF K%=13 UNTIL-1:ENDPROC
1360 IF K%>135 AND K%<140 PROCmove(NUM%)
1370 IF K%=135 PROCcopy
1380 IF K%=9 PROCentervalue
1390 IF K%=127 PROCzero
1400 IF LP%<>P% LP%=P%:PROCdisplayvalue

1420 UNTIL0
1430 :
1440 DEF PROCzero
1450 NUM%(Y%,X%)=0:ENDPROC
1460 :
1470 DEF PROCcopy
1480 IF X%=1 VDU7:ENDPROC
1490 NUM%(Y%,X%)=NUM%(Y%,X%-1)
1500 PROCshowcell(NUM%(Y%,X%))
1510 ENDPROC
1520 :
1530 DEF FNpad(N%,T)
1540 =RIGHT$(STRING$(N%,CHR$32)+STR$T,
N%)
1550 :
1560 DEF PROCmove(N%)
1570 IF K%=136 X%=X%-1
1580 IF K%=137 X%=X%+1
1590 IF K%=138 Y%=Y%+1
1600 IF K%=139 Y%=Y%-1
1610 IF X%=0 X%=12 ELSE IF X%=13 X%=1
1620 P%=((X%-1)AND12)+1
1630 IF Y%=0 Y%=N% ELSE IF Y%=N%+1 Y%=1
11640 ENDPROC
1650 :
1660 DEF PROCdisplayvalue
1670 PROCcolour(2):CLS
1680 PROCcolour(1):FORI%=1TO4
1690 PRINTTAB((I%-1)*8,0);MON$(P%+I%-1)
1700 NEXT:PROCcolour(2)
1710 FORI%=1 TO4:FORJ%=1TONUM%
1720 IF P%+I%-1=X% AND J%=Y% PROCcolou
r(1) ELSE PROCcolour(2)
1730 PRINTTAB((I%-1)*8,J%+ABS(J%>=SPLI
T%));
1740 IF NUM%(J%,P%+I%-1) A$=FNpad(8,NU
M%(J%,P%+I%-1)/100) ELSE A$=pad$
1750 PRINTA$:NEXT,
1760 PROCcolour(1)
1770 VDU28,8,31,39,28,12
1780 PRINT"Return......Exit to menu."
1790 PRINT"Tab.........Enter value."
1800 PRINT"Cursor keys.Move cell."
1810 PRINT"Copy........Copies cell.";
1820 VDU28,8,27,39,0:PROCcolour(2)
1830 ENDPROC
1840 :
1850 DEF PROCdisplaysinglevalue
1860 PROCcolour(2)
1870 PRINTTAB((X%-P%)*8,Y%+ABS(Y%>=SPL
IT%));
1880 IF NUM%(Y%,X%) A$=FNpad(8,NUM%(Y%
,X%)/100) ELSE A$=pad$
1890 PRINTA$:ENDPROC
1900 :
1910 DEF PROCshowcell(T)
1920 PROCcolour(1)
1930 PRINTTAB((X%-P%)*8,Y%+ABS(Y%>=SPL
IT%));
1940 IF T A$=FNpad(8,T/100) ELSE A$=pa
d$
1950 PRINTA$:ENDPROC
1960 :
1970 DEF PROCshift
1980 X%=X%+4
1990 IF X%=0 X%=12 ELSE IFX%>=12 X%=1
2000 P%=((X%-1)AND12)+1:LP%=0
2010 ENDPROC
```

```
 2020 :
 2030 DEF PROCentervalue
 2040 PROCcolour(1)
 2050 PRINTTAB((X%-P%)*8,Y%+ABS(Y%>=SPL
IT%))STRING$(8,CHR$32)
 2060 INPUTTAB((X%-P%)*8,Y%+ABS(Y%>=SPL
IT%))T
 2070 NUM%(Y%,X%)=T*100:ENDPROC
 2080 :
 2090 DEF PROCinit
 2100 B%=4:F%=3:PROCcolours:@%=10
 2110 OPNBAL=0:pad$="   ==   "
 2120 NUM%=24:SPLIT%=14:PROCmonths
 2130 DIM NUM%(NUM%,12),HDG$(NUM%)
 2140 FORI%=1TOSPLIT%-1
 2150 HDG$(I%)="Credit"+FNpad(2,I%)
 2160 NEXT:FORI%=SPLIT%TONUM%
 2170 HDG$(I%)="Debit"+FNpad(3,1+I%-SPL
IT%)
 2180 NEXT:@%=&1020208
 2190 DIM BCF%(13):ENDPROC
 2200 :
 2210 DEF PROCcolours
 2220 VDU19,0,B%,0,0,0
 2230 VDU19,1,F%,0,0,0
 2240 VDU23,1,0;0;0;0;
 2250 ENDPROC
 2260 :
 2270 DEF PROCerror1
 2280 ON ERROR OFF:VDU26,20,12
 2290 IF ERR<>17 REPORT:PRINT" at line
";ERL
 2300 ENDPROC
 2310 :
 2320 DEF PROCerror2
 2330 LOCAL A%:VDU26,20,12:E%=0
 2340 IF ERR=17 AND NOT INKEY-1 ENDPROC
 2350 REPORT:PRINT" at line ";ERL
 2360 PRINT"Continue? (Y/N)";
 2370 REPEAT A%=GET AND &DF
 2380 UNTIL A%=89 OR A%=78
 2390 *FX4
 2400 E%=(A%=78):ENDPROC
 2410 :
 2420 DEF PROCsave
 2430 LOCAL T%:PROCcolour(2)
 2440 VDU26,12:PROCcolour(1)
 2450 PROCspread("Data saving section."
,1)
 2460 PROCcolour(2):VDU28,0,31,39,3
 2470 PROCfile:T%=OPENOUT F$
 2480 PRINT#T%,DATE$,OPNBAL,NUM%,SPLIT%
 2490 FORI%=1 TONUM%:PRINT#T%,HDG$(I%)
 2500 FORJ%=1 TO12:PRINT#T%,NUM%(I%,J%)
 2510 NEXT,:CLOSE#T%:VDU26,20,12
 2520 ENDPROC
 2530 :
 2540 DEF PROCload
 2550 LOCAL T%:PROCcolour(2)
 2560 VDU26,12:PROCcolour(1)
 2570 PROCspread("Data loading section"
,1)
 2580 PROCcolour(2):VDU28,0,31,39,3
 2590 PROCfile:T%=OPENIN F$
 2600 IF T%=0 CLOSE#T%:PRINT''"File not
 found!"'"Return for menu.":REPEATUNTIL
GET=13:ENDPROC
 2610 INPUT#T%,DATE$,OPNBAL,NUM%,SPLIT%
 2620 DIM NUM%(NUM%,12),HDG$(NUM%)
 2630 FORI%=1 TONUM%:INPUT#T%,HDG$(I%)
 2640 FORJ%=1 TO12:INPUT#T%,NUM%(I%,J%)
 2650 NEXT,:CLOSE#T%:VDU26,20,12
 2660 ENDPROC
 2670 :
 2680 :
 2690 DEF PROCmonths
 2700 DIM MON$(12):RESTORE 1090
 2710 FORI%=1 TO12:READ A$
 2720 MON$(I%)="  ="+A$+"= ":NEXT
 2730 ENDPROC
 2740 :
 2750 DEF PROCtab
 2760 PROCcolour(2):VDU12,28,0,28,7,2
 2770 PROCcolour(1):CLS
 2780 FORI%=1 TOSPLIT%-1:PRINTHDG$(I%);
 2790 NEXT:PRINT:FORI%=SPLIT%TONUM%
 2800 PRINTHDG$(I%);:NEXT
 2810 PRINTTAB(0,26)"Balance";
 2820 PROCcolour(2)
 2830 VDU28,8,31,39,0,12,26
 2840 PROCcolour(1)
 2850 PRINTTAB(3,31)"Date:";
 2860 VDU28,8,31,39,0
 2870 PRINTTAB(0,31)DATE$;TAB(17,31)"Sp
ace/Return";
 2880 PROCcolour(2)
 2890 X%=1:P%=1:BCF%(0)=OPNBAL
 2900 FORI%=1 TO12
 2910 BCF%(I%)=BCF%(I%-1)
 2920 FORJ%=1TOSPLIT%-1
 2930 BCF%(I%)=BCF%(I%)+NUM%(J%,I%)
 2940 NEXT:FORJ%=SPLIT%TONUM%
 2950 BCF%(I%)=BCF%(I%)-NUM%(J%,I%)
 2960 NEXT,:REPEAT:PROCtabbed:A%=GET
 2970 IF A%=32 PROCshift
 2980 UNTILA%=13:ENDPROC
 2990 :
 3000 DEF PROCtabbed
 3010 FORI%=1TO4
 3020 PROCcolour(1+(I%AND1)):PRINTTAB((
I%-1)*8,0)MON$(I%-1+P%)
 3030 PROCcolour(2-(I%AND1)):PRINTTAB((
I%-1)*8,1);:T=BCF%(I%-2+P%)/100
 3040 IF T=0 PRINTpad$ ELSE PRINTT
 3050 NEXT:PROCcolour(2)
 3060 FORI%=1TO4
 3070 FORJ%=1TOSPLIT%-1
 3080 PRINTTAB((I%-1)*8,J%+1);:T=NUM%(J
%,I%-1+P%)/100
 3090 IF T=0 PRINTpad$ ELSE PRINTT
 3100 NEXT:FORJ%=SPLIT%TONUM%
```

»

```
 3110 PRINTTAB((I%-1)*8,J%+2);:T=NUM%(J
%,I%+P%-1)/100
 3120 IF T=0 PRINTpad$ ELSE PRINTT
 3130 NEXT:PROCcolour(1)
 3140 PRINTTAB((I%-1)*8,28);:T=BCF%(I%-
1+P%)/100
 3150 IF T=0 PRINTpad$ ELSE PRINTT
 3160 PROCcolour(2):NEXT:ENDPROC
 3170 :
 3180 DEF PROCos
 3190 VDU26,12
 3200 PROCcntr("OS Command.",1)
 3210 INPUT"*"$&C00:X%=0:Y%=&C
 3220 VDU14:CALL&FFF7
 3230 PRINT"Press SPACE to continue:";
 3240 REPEAT UNTIL GET=32:CLS
 3250 ENDPROC
 3260 :
 3270 DEF PROCdate
 3280 PRINT'''':REPEAT
 3290 INPUTTAB(15)"Date :"DATE$
 3300 UNTIL LENDATE$<11:ENDPROC
 3310 :
 3320 DEF PROCtitles
 3330 *FX4 1
 3340 PROCcolour(2):CLS
 3350 PROCcolour(1):P%=1
 3360 VDU28,0,31,39,28,12
 3370 PRINTTAB(0,29)"Cursor keys.Move p
ointer"
 3380 PRINT"Return......Exit to menu"
 3390 PRINT"Press Tab, Enter String, th
en Press     Return.";
 3400 VDU28,0,27,39,0:REPEAT
 3410 PROCcolour(1)
 3420 PRINTTAB(0,0)"Credit Headings:"
 3430 PROCcolour(2)
 3440 FORI%=1TOSPLIT%-1
 3450 PRINTHDG$(I%):NEXT
 3460 PROCcolour(1)
 3470 PRINT"Debit Headings:"
 3480 PROCcolour(2)
 3490 FORI%=SPLIT%TONUM%
 3500 PRINTHDG$(I%):NEXT
 3510 PRINTTAB(20,P%-(P%>=SPLIT%));"<==
="
 3520 PROCcolour(1)
 3530 PRINTTAB(0,P%-(P%>=SPLIT%))HDG$(P
%)
 3540 PROCcolour(2):A%=GET
 3550 *FX15,1
 3560 PRINTTAB(20,P%-(P%>=SPLIT%));SPC4
 3570 IF A%=9 PROCenterthisheading
 3580 IF A%=138 P%=P%+1
 3590 IF A%=139 P%=P%-1
 3600 IF P%=0 P%=NUM%
 3610 IF P%>NUM% P%=1
 3620 UNTIL A%=13:*FX4
 3630 ENDPROC
 3640 :
 3650 DEF PROCenterthisheading
 3660 VDU28,20,1+P%-(P%>=SPLIT%),39,P%-
(P%>=SPLIT%),12
 3670 REPEAT INPUT HDG$(P%)
 3680 UNTIL LEN HDG$(P%)<9
 3690 HDG$(P%)=LEFT$(HDG$(P%)+STRING$(8
,CHR$32),8)
 3700 VDU12,26:ENDPROC
 3710 :
 3720 DEF PROCquit
 3730 VDU26,20,12:END
 3740 :
 3750 DEF PROCsummary
 3760 RESTORE 1090:PROCcolour(2)
 3770 VDU26,12:PROCcolour(1)
 3780 PRINT"Balances at end"'"of  each
month."'
 3790 PROCcolour(2):BCF%(0)=OPNBAL
 3800 FORI%=1TO12
 3810 BCF%(I%)=BCF%(I%-1)
 3820 FORJ%=1TOSPLIT%-1
 3830 BCF%(I%)=BCF%(I%)+NUM%(J%,I%)
 3840 NEXT:FORJ%=SPLIT%TONUM%
 3850 BCF%(I%)=BCF%(I%)-NUM%(J%,I%)
 3860 NEXT,:FORI%=1TO12
 3870 PRINTTAB(2);:PROCcolour(1)
 3880 READ A$:PRINTA$
 3890 PROCcolour(2)
 3900 PRINTTAB(7)BCF%(I%)/100:NEXT
 3910 PROCcolour(1)
 3920 PRINTTAB(18,1)"Income/Outcome Sum
mary"
 3930 PROCcolour(2)
 3940 FORI%=1TONUM%
 3950 PRINTTAB(20,I%+1)HDG$(I%);
 3960 T%=0:FORJ%=1TO12
 3970 T%=T%+NUM%(I%,J%):NEXT
 3980 IF T% PRINTT%/100 ELSE PRINTpad$
 3990 NEXT:PROCcolour(1)
 4000 PRINTTAB(0,29)"Return to continue
."
 4010 REPEAT UNTIL GET=13:ENDPROC
 4020 :
 4030 DEF PROCfile
 4040 REPEAT
 4050 INPUT"Filename :"F$
 4060 UNTIL LENF$<11:ENDPROC
 4070 DATA Load Data, Enter Headings,En
ter/Amend Data,Tabulation,Summary Table
,Save Data,* Command,Finish
```

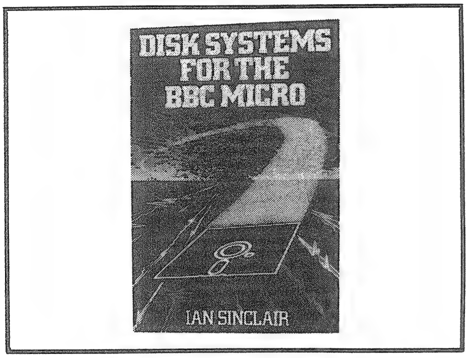
£ £

£ £

TWO BOOKS FOR DISC USERS

Reviewed by Sheridan Williams

Disk Systems for the BBC Micro by Ian Sinclair published by Granada at £6.95.
ISBN Ø-246-12325-7



Ian Sinclair has an easy style and I liked this book very much because of it. This is partly due to Ian's depth of knowledge about the Beeb, and also because with over 4Ø previous books on electronics and computing, he has a great deal of experience in writing readable material.

If you have not yet decided on a disc system for your Beeb, then this book is much more useful than Tony Latham's because, as the book's title would suggests, several systems are covered. Sinclair describes not only the Acorn DFS but also Watford's and the Pace/AMCOM DFS.

The later chapters are devoted to filing. I was particularly interested in this aspect because I have not yet read a book that covers files in anything but a superficial way. Ian's description of sequential files is clear and understandable. I was also pleased to see some time devoted to random files, and several examples were given explaining their use. To finish the section on files Ian mentions (albeit briefly) Index Sequential files (ISAM files). It is strange that Sinclair seems to miss the point about ISAM files used in his example, by suggesting a serial search through the index file to find the start address for the required record, when if he had stored the start address rather than the record length for each record he would have achieved the same ends without having to read the whole index file serially.

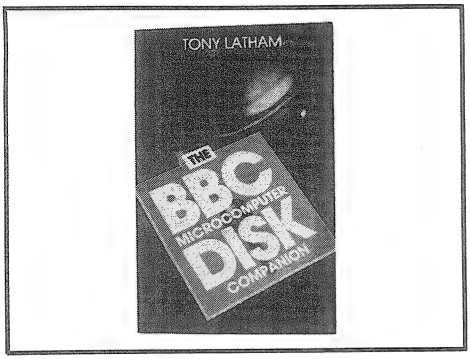
On page 11 Sinclair is describing the various configurations of double/single sided, 4Ø/8Ø track, double/single sided drives, and says "It's not advisable to use double sided discs if you have a single sided drive, because a single sided drive will not offer the same support to the opposite side." I cannot understand this as there is a felt pressure pad opposite the head, and the discs themselves are of identical construction. For many people who intend to buy a double sided drive at a later date but currently only have a single sided drive it is most sensible to buy DS discs right from the start, so that they can be used on both sides when the DS drive is later acquired.

Topics covered in the book are:-
 About discs and disc systems
 The various disc filing systems and their features.
 *SAVE *LOAD *DUMP *LIST *TYPE *OPT auto-booting.
 Text files and their problems (including Wordwise).
 Disc Utilities and how to use them.
 Basic filing techniques.
 DFS commands.

In conclusion this book is full of good ideas, and contains very few errors, though one consistent error is the use of filenames longer than that permitted on the Acorn DFS.

The BBC Microcomputer Disk Companion by Tony Latham published by Prentice Hall International at £7.95.
ISBN Ø-13-Ø69311-1

This book is nothing like as easy to read as the one by Ian Sinclair. I found many of the descriptions fundamental to the understanding of discs hard to follow, and would suggest that the beginner needs a much clearer



approach. This considerably devalues the book for those who use it for learning about their Acorn DFS.

The copyright on this book is 1983, and it is obvious in several places that the book is not as up to date as Sinclair's. In fact Tony told me he was working on this book in February 1983. The book is also written for the American market with the American spellings of 'color' 'center' 'endeavor' etc.

This book is really aimed at those with the Acorn DFS, and no others are described.

Much of the book is based around three programs given in the book - a disc formatter, verifier, and a word processor. The formatter is rather a waste of space because all suppliers of discs systems for the Beeb now supply a formatting program. Latham spends a fair amount of text explaining how these programs work. These programs are hardly the type of program that the beginner would tackle.

The chapter on file handling is in my view poor, and contains misleading statements like "New Basic requires OPENUP instead of OPENIN." The truth is that Basic II has both OPENIN and OPENUP, each having a different use, however to maintain compatibility between Basic I and Basic II you should only use OPENUP on machines that have Basic II.

Latham's description of random files in Basic is virtually non-existent and is mostly described using assembly language instructions. This will probably put the beginner off altogether.

SUMMARY

For me there is no competition between the books, Sinclair's is far better value for money (and it's £1 cheaper anyway). The main justification for buying Latham's book would be if you were familiar with disc systems but wanted descriptions of the OS statements OSFIND, OSARGS, OSFILE etc, in which case the Advanced User Guide provides much more detail anyway.

I still maintain that there is a need for a complete book devoted to file handling, totally separate from books on disc systems. I await such a book with interest.

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

SEIKOSHA POUND PATCH - V.Haworth

The following program will correctly print a pound sign on a Seikosha printer connected to a BBC micro. First set the printer to the USA setting, run the program, and press Break. After this all occurrences of '£' will be printed correctly.

This program changes the contents of the buffer insert vector located at &22A and &22B to 'point to itself'. The character &60 (BBC pound) is searched for and changed to &B9 (Seikosha pound), providing the printer is enabled.

```
10 FOR I%=0 TO 2 STEP 2:P%=&90
20 [OPT I%
30 CMP #&60   \ Is character
40 BNE exit   \ a '£'.
50 CPX #3     \ Printer buffer
60 BNE exit   \ is 3.
70 LDA #&B9   \ Seikosha '£'.
80 .exit
90 JMP &E4B3:]
100 NEXT
110 ?&22A=&90:?&22B=00
120 *K.10 ?&22A=&90:?&22B=0:VDU 12|M
```

Tested on Basic I & II and O.S. 1.2

INDEXING PROGRAMS ON CASSETTE

by D. J. Pilling

This is a very handy utility for all cassette users as it not only provides an index of programs that you have saved on a cassette but it also works out the counter settings as well for speedy access.

One of the problems with keeping several programs on a cassette is that of accessing the programs when you want to use them. Unless you know fairly accurately where a program is saved, much time can be wasted just finding where the start of the program is. Most cassette players used with computers have a counter which can be very useful in identifying the start of any given program. We now go one step further and automate the whole process of cataloguing a cassette of programs with the utility listed here.

The indexer utility will automatically read through a complete cassette of programs and produce an index giving for each program its name, its length in blocks and in bytes, starting time in seconds and starting counter reading. The indexer also has some capability to cope with programs which contain recording errors (i.e. which would not load under normal circumstances). If such a program is found, its length in bytes is entered as a blank, providing an indication of its bad state.

The rate at which the counter rotates varies, sometimes significantly, from one cassette recorder to another. For this reason, the indexer program contains a special calibration routine to enable it to work reasonably accurately with any cassette recorder. Once the program has been calibrated in this way (in effect by setting the values of two variables R and D) the indexer will automatically record the correct counter readings as it reads through a cassette of programs. In addition, the indexer program can also be used in manual mode, with you entering the correct counter value at the start of each program on a cassette. The indexer program contains full instructions to guide you through whichever option you choose.

USING THE INDEXER PROGRAM

Before using the indexer program you should make sure you have at least one copy securely saved. This is important as the indexer program modifies itself when it runs, leaving you only with the cassette index at the end of its run. The program details the three choices open to you. These are:

1. Manual indexing of cassette.
2. Automatic indexing of cassette.
3. Calibration and indexing.

This program was originally designed for cassette recorders that had remote motor control. In order that the program may also be used with other cassette recorders, we have provided a separate calibration program to use first in this case. See the end of the article for further details on this.

CALIBRATION

This section allows the values of R and D for your cassette recorder to be calculated precisely. Once selected, the tape will start running, and you should press the space bar when the prompt is displayed. You will then be asked to enter the counter reading. The program will continue to read through the tape, indexing the programs as it finds them. After a reasonable interval (to get a good average reading) press the space bar a second time and enter the new counter reading. The program then calculates the correct values of the constants R and D before continuing to index the programs on the tape as described below.

AUTOMATIC INDEXING

With this option, the program reads through the tape, and notes down each file, its length, where on the tape it starts, and at what tape counter value. When you are satisfied that all the files have been included, press Escape ONCE ONLY. The program will then modify itself, leaving you with a small program that when run, lists out the files on the tape, in order, with their associated statistics. You can save this program if required, and also, if available, list out the index on a printer by including Ctrl-B when you type the RUN command.

MANUAL INDEXING

With this option, you are required to enter the value of the tape counter at the start of each program, but aside form that, it is functionally equivalent to the automatic indexing option. Because of the way in which this option is performed, only cassette users with motor control will be able to use this option correctly.

```
    1 REM PROGRAM TAPE INDEXER
    2 REM AUTHOR D.J. PILLING
    3 REM VERSION B1.2
    4 REM BEEBUG APRIL 1984
    5 REM PROGRAM SUBJECT TO COPYRIGHT
    6 :
   50 MODE7:@%=&20406
   60 PROCINS
   70 *OPT1,1
   80 DIMN$(100),T$(100),C$(100)
   90 ONERROR GOTO580
   95 PROCTITLE
  100 PRINT'''CHR$134"(1)"CHR$135"MANU
AL"'CHR$134"(2)"CHR$135"AUTO"'CHR$134"(
3)"CHR$135"CALIBRATE"'''CHR$130"SELECT
TAPE COUNTER MODE:";:INPUT""C%
  110 IFC%<1 OR C%>3 GOTO100
  120 TS%=0:TF%=0:N%=0
  130 REM =========================
  140 REM DEFAULT R AND D
  150     R=.2779:D=.0003
  160 REM =========================
  170 CLS:TIME=TF%
  180 X%=OPENIN("")
  190 TS%=TIME:N%=N%+1:PROCCAL:TIME=TS%
  200 REPEAT Z%=BGET# X%:UNTIL EOF#X%
  210 CLOSE #X%
  220 TF%=TIME
  230 PROCST
  240 GOTO170
  250 IF C%>2 PROCCX
  260 HIMEM=HIMEM-8000:P%=HIMEM
  270 PROCDS:P%=HIMEM
  280 L%=6000
  290 F$=STR$L%+"D."+STR$N%+CHR$13+"G.3
10"+CHR$13
  300 PROCWR(F$)
  310 L%=6010:I%=1
  320 F$=STR$L%+"D."+FNSS+CHR$13+"G.340
"+CHR$13
  330 PROCWR(F$)
  340 L%=L%+5
  350 F$=STR$L%+"D."+FNSS+","+FNSS+CHR$
13+"G.370"+CHR$13
  360 PROCWR(F$)
  370 L%=L%+10:I%=I%+1
  380 IFI%>N% ELSE GOTO320
  390 F$="DELETE0,1500"+CHR$13
  400 CLS:PRINT''"I'M YOUR INDEX"''"SAV
E ME"
  410 PRINT'''"(The program in the comp
uter is now an"'"index of the tape; whi
ch you may run"'"and save.)"''
  420 PROCWR(F$)
  430 END
  440 DEFPROCWR(A$)
  450 FORJ%=1TOLENA$
  460 A%=&8A:X%=0:Y%=ASC(MID$(A$,J%,1))
  470 CALL&FFF4
  480 NEXT:END
  490 DEFPROCDS
  500 FORI%=1TON%
  510 $P%=N$(I%):P%=P%+LENN$(I%)+1
  520 $P%=T$(I%):P%=P%+LENT$(I%)+1
  530 $P%=C$(I%):P%=P%+LENC$(I%)+1
  540 NEXT:ENDPROC
  550 DEFFNSS
  560 A$=$P%:P%=P%+LEN$P%+1
  570 =A$
  580 IF ERR=17 AND ERL=180 GOTO250
  590 IF ERR=17 AND ERL=200 N%=N%-1:GOT
O250
  600 IF ERR>200 TF%=TIME:PRINT:PROCST:
GOTO170
  610 ONERROROFF:REPORT:PRINT" at line
";ERL:END
  620 DEFPROCST
  630 F$=FNS(0,VPOS-1)
  640 L$=RIGHT$(F$,4)
  650 B$=MID$(F$,12,2)
  660 F$=LEFT$(F$,10)
  670 IFMID$(F$,10,1)<>" " GOTO700
  680 REPEAT F$=LEFT$(F$,LENF$-1)
  690 UNTIL MID$(F$,LENF$,1)<>" "
  700 N$(N%)=""""+F$+B$+L$+""""
  710 T$(N%)=STR$(TS%DIV100)
  720 C$(N%)=C$
  730 ENDPROC
  740 DEFFNS(X,Y)
  750 A$="":FORI%=0TO17:A$=A$+CHR$(?(HI
MEM+X+I%+40*Y)):NEXT
  760 =A$
  770 DEFPROCCAL
  780 *FX21,0
  790 Y%=VPOS
  800 IF C%=1:VDU7,7:INPUTTAB(0,Y%+1)"N
UMBER ON TAPE COUNTER  "C$:IFC$=""GOTO8
00
  810 IF C%=2 C$=FNTC(TS%DIV100)
  820 IF C%>2 C$="":PROCCALS
  830 PRINTTAB(0,Y%);
  840 ENDPROC
```

»

```
  850 DEFPROCCALS
  860 VDU7,7
  870 PRINTTAB(0,Y%+1)"TO GO FOR CALIBR
ATION PRESS SPACE BAR":IK=INKEY(200)
  880 IF IK<>32 PRINTTAB(0,Y%+1)SPC(39)
:ENDPROC
  890 INPUT'"INPUT NUMBER ON TAPE COUNT
ER  "N1%
  900 IFN1%=0 PRINT"ERROR ZERO ENTERED"
:IK=INKEY(200):ENDPROC
  910 IF C%=3 N2%=N1%:T2%=TS%DIV100:C%=
4:ENDPROC
  920 IF N1%=N2% PRINT"ERROR SAME NUMBE
R ENTERED TWICE":IK=INKEY(200):GOTO890
  930 T1%=TS%DIV100:PROCCDR
  940 IFD=0 PRINT"ERROR INSUFFICIENT AC
CURACY"'"TRY AGAIN LATER":IK=INKEY(200)
:ENDPROC
  950 PROCCX:C%=2:C$=FNTC(TS%DIV100)
  960 PRINT'"CALIBRATION COMPLETED"'
  970 PRINT"YOUR D VALUE = ";D
  980 PRINT"YOUR R VALUE = ";R
  990 PRINT'"(Retain all digits. Next t
ime you LOAD"'"the program enter R and
D in Line 150.)"
 1000 PRINT'CHR$130"ANY KEY TO CONTINUE
"
 1010 IK=GET
 1020 ENDPROC
 1030 DEFPROCCDR
 1040 D=(T1%/N1%-T2%/N2%)/PI/(N1%-N2%)
 1050 R=(T1%/N1%/(N1%-1)-T2%/N2%/(N2%-1
))/PI/2/(N2%-N1%)*(N1%-1)*(N2%-1)
 1060 ENDPROC
 1070 DEFFNTC(ST%)
 1080 =STR$(INT((D-2*R+SQR((2*R-D)^2+4*
ST%*D/PI))/2/D+0.5))
 1090 DEFPROCCX
 1100 FOR J%=1TON%:C$(J%)=FNTC(VAL(T$(J
%))):NEXT:ENDPROC
 1110 DEFPROCINS
 1111 PROCTITLE
 1120 VDU7:PRINT''CHR$129;"WARNING THIS
 PROGRAM IS SELF"'CHR$129"MODIFYING. BE
FORE PROCEEDING"'CHR$129;"MAKE SURE YOU
 HAVE A COPY."''TAB(17)CHR$130"ANY KEY
TO START";:IK=GET:CLS:*FX21,0
 1130 PROCTITLE:PRINT'CHR$134;"HOW TO U
SE THE PROGRAM"'"Put the tape you want
an index of in"'"the recorder. Zero the
 tape counter."'"You now have three opt
ions:"
 1140 PRINTCHR$134;"(1) MANUAL"'"Enter
the numbers on the counter"'"at the sta
rt of each program as"'"prompted."
 1150 PRINTCHR$134"(2) AUTO"'"Let the p
rogram calculate the"'"tape counter rea
dings; using default"'"R and D values p
reviously determined"'"for your recorde
r and type of tape"'"and entered in lin
e 150."
 1160 PRINTCHR$134"(3) CALIBRATE"'"Inpu
t two tape counter readings to"'"calibr
ate the program for your"'"recorder and
 then proceed as in (2)."
 1180 PRINT''TAB(17)CHR$130"SHIFT TO ST
ART";
 1190 REPEAT UNTILINKEY-1:*FX21,0
 1200 ENDPROC
 1210 :
 1220 DEF PROCTITLE
 1230 CLS
 1240 PRINTTAB(10)CHR$129;CHR$157;CHR$1
31;"TAPE INDEXER  ";CHR$156
 1250 ENDPROC
 1260 :
 5000 CLS:PRINTTAB(13)CHR$129CHR$157CHR
$131"TAPE INDEX  "CHR$156
 5010 Y%=1:RESTORE6000:READN%:PRINTCHR$
134"NAME"TAB(12)"B"TAB(16)"L"TAB(22)"TI
ME"TAB(30)"TAPE"
 5020 FORI%=1TON%
 5030 READN$,T%,C%
 5040 L$=RIGHT$(N$,4):B$=MID$(N$,LENN$-
5,2):N$=LEFT$(N$,LENN$-6)
 5050 PRINTN$TAB(12)B$TAB(16)L$TAB(23);
T%TAB(31);C%
 5060 NEXT
```

Note: The program above uses some rather unconventional coding techniques, which also accounts for the fact that the line numbers do not follow our usual pattern. Do not renumber this program.

MANUAL CALIBRATION

Those whose cassette recorders do not have motor control should use this separate program to calibrate the system before using option 2 in the main program already described. When you run the calibration program, switch your cassette recorder to 'play' and note the counter reading on each of the two occasions that you hear a bleep. The program will then ask you to enter these two readings which are used to calculate the values of R and D which you should then include in the main program at line 150.

```
   10 REM PROGRAM NO MOTOR CALIBRATE
   20 REM AUTHORS D.J. PILLING/ D. FELL
   30 REM VERSION B1.1
   40 REM BEEBUG APRIL 1984
   50 REM PROGRAM SUBJECT TO COPYRIGHT
   60 :
  100 *OPT1 1
  110 MODE 7
  120 T2%=0
  130 PRINTCHR$130;"MANUAL CALIBRATION.
"
```

```
  140 PRINT'''"NOTE THE COUNTER VALUES O
N THE TONES."
  150 X%=OPENIN""
  160 TIME=0
  170 SOUND17,-15,100,10
  180 REPEAT B%=BGET#X%:UNTILEOF#X%
  190 SOUND17,-15,100,10
  200 T1%=TIME DIV 100
  210 CLOSE#X%
  220 INPUT"FIRST COUNTER VALUE "N2%'"S
ECOND COUNTER VALUE "N1%
  230 PROCCDR
  240 PRINT"D VALUE IS "D'"R VALUE IS "
R
  250 PRINT"NOTE THESE AND USE AT LINE
150 OFTHE"'"MAIN PROGRAM. ALL DIGITS AR
E SIGNIFICANT"
  260 END
  270 :
 1000 DEFPROCCDR
 1010 D=(T1%/N1%-T2%/N2%)/PI/(N1%-N2%)
 1020 R=(T1%/N1%/(N1%-1)-T2%/N2%/(N2%-1
))/PI/2/(N2%-N1%)*(N1%-1)*(N2%-1)
 1030 ENDPROC
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

DIRECT MEMORY ACCESS ACROSS THE TUBE

A relatively unknown fact about the BBC micro, is that you can read from, and write to, specific memory locations in the Beeb without worrying about Tube incompatibility. This is achieved by using OSWORD with A=&5 (for read) and A=&6 (for write). The X and Y registers hold an address pointing to four bytes of memory containing a 32 bit address i.e. XY+0=LSB of address, XY+3=MSB of address. XY+4 contains the byte to be written, or the byte just read. Usually the most significant 16 bits are unused, but, setting these to &FFFF will specifically select the BBC micro's memory range.

'NEXT,' EFFECT WITH LISTO7 - M.Robinson

In BEEBUG Vol.1 No.8 it was pointed out that a series of FOR..NEXT loops can be terminated with commas e.g. NEXT,,. However these commas are not recognised by the LISTO7 option, used to format listings by indenting by two spaces for each nested loop. Loops terminated by a comma will not be indented by an extra two spaces for each NEXT that has been replaced by a comma.

This also applies if the controlling variable of the loop is included e.g. NEXT A,B,C. The actual keyword 'NEXT' must be included to cancel the indentation e.g. NEXT:NEXT:NEXT.

TELETEXT DOWNLOADER CLASH - K.Wood

In the downloader program on page 704 of Teletext, lines 32715, 32716 and 32717 usually contain abbreviations for the commands PAGE (P.), EXEC (E.) and BASIC (B.). These commands are passed to the operating system for interpretation. For example *PAGE, is a command recognised by the Teletext Filing System (TFS). However, some of these abbreviations may be intercepted by other ROMs, in particular Toolkit which will interpret P. as *PACK. We therefore advise you to ammend the commands in the downloader program to the full versions, as outlined above, before you save a copy and use it to download other programs.

DIFERENCES BETWEEN THE AMCOM DFS AND ACORN DFS

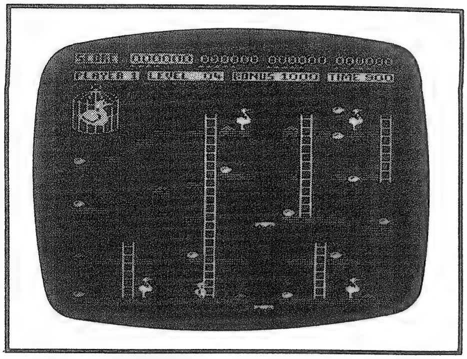
Unlike the Acorn DFS, the AMCOM DFS will not accept '*' as the wild card for any directory, but must have '#' as the wild card. i.e. *INFO *.* will only act on files in the current directory, but *INFO #.* will give information on all files.

If the Acorn DFS comes across a disc with a coloured title, it will read it normally, but an AMCOM DFS will crash the Beeb because it uses the top bit of the first byte on the disc title to signify ordinary or extended mode.

Pace have told us that the new version of their DFS allows PAGE to be set as low as &1100 providing you do not use data files, or &1200 if you do. The default seting for PAGE is &1500. In addition, it does not use any of the other internal buffers, unlike the older versions of this DFS.

LATEST GAMES REVIEWED

by David Fell and Mike Beasley



Title	: Chuckie Egg
Supplier	: A & F Software
Price	: £7.90
Rating	: ****

Chuckie Egg is one of those addictive games that you could happily play all day, given nothing else to do. The scenario is very basic; you are a rather peculiar looking bird called Chuckie Egg, and you have to walk and fly around a complex of ladders, floors, and (later on) elevators, collecting eggs. Each round is completed when all of the eggs have been gathered, and you can gain extra points by eating the piles of grain that are also present. There is a bonus for completing the round as quickly as possible, and hazards are provided in the form of stork-like birds, which eat the piles of grain, and kill you if you get too close, with more features appearing as the game advances.

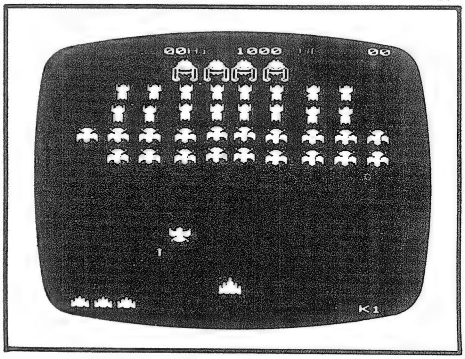
The game gets increasingly harder as you progress through different levels, and the graphics, although perhaps a little simple, are quite pleasing, as are the sound effects.

Title	: Zalaga
Supplier	: Aardvark
Price	: £6.90
Rating	: ****

Zalaga is a very successful variant on the "blast 'em out of the sky" theme. The aliens swoop in from various directions to form themselves into a Galaxians type layout at the top of the screen. Once this is complete, they swoop down on you, while at the same time you are trying to shoot as many as possible of them. A screen is completed when all the aliens are destroyed. Interspersed between these standard screens (which grow steadily faster and more vicious) are the Challenge Stages, where the aliens fly around in set patterns, the object being to shoot as many as possible while they remain on screen. It is also occasionally possible to summon up a second fire base to act in concert with the current one, doubling the fire-power (and target area).

One very good point about this game is that it does not become tedious once you get good at it, as games addicts who've tried it will testify. The sound and graphics are very good, the latter being both fast and smooth, even if occasionally the aliens do disappear for a short time owing to a slight timing fault. This problem is not severe, and is hardly noticable once you get engrossed in the game. The handling is very good, the controls are simple, and a nice touch is the scrolling backdrop of stars.

Overall, this is a superb game, probably the best game of its genre with the inevitable exclusion of Acornsoft's Planetoid.



Tested on Basic I & II and O.S. 1.2

MOVING TEXT DISPLAY(16K)

by R. R. Hull

The procedure described here provides an interesting visual way of displaying up to 255 characters on a single line, by scrolling the text sideways across the screen.

The effect of this procedure is to present a line of text which moves horizontally across the screen from right to left as though being viewed through a window. The procedure is quite versatile and will allow you to position the window anywhere on the screen and determine its length. This moving text display is similar in appearance to that seen on the outside of buildings in some large cities.

The procedure can be used in any mode and is defined as follows:

```
PROCscroll(col%,X%,Y%,length%,T%,D%,C$,
text$)
```

The various parameters used in the procedure are explained in the following table:

col%	Sets the colour of the text using COLOUR.
X%	Sets the left hand position of the text window.
Y%	Sets the vertical position of the text line.
length%	The length of the window for scrolling the text.
T%	Sets the scrolling rate.
D%	Determines the number of lines printed (set to 1 for double height text in mode 7, otherwise set to Ø).
C$	Any special codes to be inserted (e.g. double height in Mode 7).
text$	The text to be printed (up to 255 characters).

The parameter T% (in hundreths of a second) is the time for the text message to move one character.

The use of this procedure is demonstrated by the accompanying program. This scrolls a suitable message across line 15 on the screen in Mode 1, using a window 5 characters in from the left hand side and 3Ø characters long. The message is repeated continually, by placing the procedure call in a loop, until the space bar is pressed.

The procedure will work with strings up to a maximum length of 255 characters. Larger pieces of text can be displayed by calling the procedure several times and inserting the new section of text into the parameter text$. In Mode 7, teletext control codes can also be inserted into the text (provision for this has been included in the program at line 14Ø). To set up red, double height text in Mode 7, lines 1ØØ, 14Ø and 18Ø need to be changed to:

```
 1ØØ MODE 7
 14Ø sc$=CHR$129+CHR$141
 18Ø PROCscroll(2,Ø,12,4Ø,2Ø,1,sc$,T$)
```

The window for the scrolling text is set within the procedure using the VDU28 command, and on exit from the procedure text and graphics windows are restored to the full screen area using VDU26. Make sure that the values for the parameters X%, Y% and length% are appropriate to the mode (and text display) in use.

Although very versatile as written, the procedure could easily be simplified for use in a specific program. For example, the parameter C$ is only relevant in Mode 7, and could be deleted together with any other references to C$ in the procedure when used in other modes. The same is true for the parameter D% and its use in line 1Ø6Ø.

```
 1Ø REM SCROLL
 2Ø REM Version B1.3
 3Ø REM Author R.R.Hull
 4Ø REM BEEBUG April 1984
 5Ø REM Program subject to copyright
 6Ø:
1ØØ MODE 1
11Ø ON ERROR GOTO 22Ø
12Ø VDU 23,1,Ø;Ø;Ø;Ø;
```

»

```
  13Ø VDU 23,244,6Ø,126,219,255,126,6Ø,
36,66
  14Ø sc$=CHR$(Ø)
  15Ø PRINT TAB(1,1)"Example";TAB(2,2)"
of";TAB(3,3)"scrolling";TAB(4,4)"text"
  16Ø T$="You can use this procedure in
 any mode, providing you correctly set
up the parameters as described in the t
ext. The text can also include colour,
graphic and control characters "+CHR$17
+CHR$3+CHR$244+" like that!........"
  17Ø REPEAT
  18Ø PROCscroll(2,5,15,3Ø,15,Ø,sc$,T$)
  19Ø UNTIL INKEY(Ø)=32
  2ØØ END
  21Ø:
  22Ø ON ERROR OFF:MODE 7
  23Ø REPORT:PRINT" at line ";ERL
  24Ø END
  25Ø:
 1ØØØ DEF PROCscroll(col%,X%,Y%,length%
,T%,D%,C$,text$):LOCAL I%,line$
 1Ø1Ø VDU28,X%,31,X%+length%,Ø
 1Ø2Ø COLOUR col%
 1Ø3Ø FOR I%=1 TO LEN(text$)+length%
 1Ø4Ø IF I%<length% line$=STRING$(lengt
h%-I%,CHR$32)+MID$(text$,1,I%) ELSE lin
e$=MID$(text$,I%-length%+1,length%)
 1Ø5Ø PRINTTAB(Ø,Y%)C$;line$;CHR$32
 1Ø6Ø IF D% PRINTTAB(Ø,Y%+1)C$;line$;CH
R$32
 1Ø7Ø TIME=Ø:REPEAT UNTIL TIME>=T%
 1Ø8Ø NEXT I%
 1Ø9Ø ENDPROC
```

Tested on Basic I & II and O.S. 1·2

APRIL FOOL by Hugh D. Brown-Smith

The short program presented here will allow you to respond to all those cruel jokes played on you by your friends on April 1st. The program need only be entered into their machine and run, the effect being that whenever the user makes a mistake the computer will say so, in no uncertain terms. In other respects the computer should behave normally; and the 'enhanced' error reporting can only be removed by switching off the machine. Break, and even Control Break do not clear it!

To use the program simply type it in and run it to assemble the machine code routine. It remains resident in the machine at &AØØ so should not interfere with other programs. The program is written to run on Basic 1 or 2, but with Basic 2 could be shortened considerably. You can easily change the message in the Basic program but remember that when displayed, the micro will automatically add the word 'Mistake', or other standard error message, at the end.

```
   1Ø REM APRIL FOOL
   2Ø REM Version B1.9
   3Ø REM Hugh D.Brown-
Smith & Mick Beasley
   4Ø REM BEEBUG April84
   5Ø REM Copyright
   6Ø :
  1ØØ code%=&AØØ
  11Ø FORX=ØTO2STEP2
  12Ø P%=code%
  13Ø [OPTX
  14Ø BCC setvectors
  15Ø RTS
  16Ø .setvectors LDA#Ø:
STA &84
  17Ø LDA &21Ø
  18Ø STA &82
  19Ø LDA &211
  2ØØ STA &83
  21Ø LDA #(start MOD 25
6)
  22Ø STA &21Ø
  23Ø LDA #(start DIV 25
6)
  24Ø STA &211
  25Ø RTS
  26Ø .joke
  27Ø LDA &2Ø2
  28Ø STA &8Ø
  29Ø LDA &2Ø3
  3ØØ STA &81
  31Ø LDA  #(aprilfool M
OD 256)
  32Ø STA &2Ø2
  33Ø LDA  #(aprilfool D
IV 256)
  34Ø STA &2Ø3
  35Ø RTS
  36Ø .aprilfool
  37Ø JMP april1
  38Ø ]
  39Ø $P%="April Fool!"
  4ØØ P%=P%+LEN($P%)+1
  41Ø [OPTX
  42Ø .april1
  43Ø LDX #2
  44Ø .april2
  45Ø LDA aprilfool,X
  46Ø CMP #&ØD
  47Ø BEQ april3
  48Ø JSR &FFE3
  49Ø INX
  5ØØ JMP april2
  51Ø .april3
  52Ø JMP (&8Ø)
  53Ø .start
  54Ø LDX &84
  55Ø INC &84
  56Ø LDA call,X
  57Ø CMP #32
  58Ø BNE send
  59Ø LDA &82
  6ØØ STA &21Ø
  61Ø LDA &83
  62Ø STA &211
  63Ø LDA #&ØD
  64Ø .send CLC:RTS
  65Ø .call
  66Ø ]
  67Ø NEXT
  68Ø $call=CHR$11+CHR$1
52+CHR$9+CHR$127+"CALL&A
1C "
  69Ø CALL joke
  7ØØ  *FX247,76
  71Ø  *FX248,ØØ
  72Ø  *FX249,1Ø
  73Ø END
```

Tested on Basic I & II and O.S. 1.2

TESTING OUT YOUR MICRO (Part 2) – THE VIDEO ULA

by Hugh D. Brown-Smith

Hugh Brown-Smith continues his series of articles to help you test out your own micro when you suspect a fault. This month he investigates the video circuitry of the BBC micro.

In this, the second part of this series I am going to deal with the video section of the BBC micro. This is handled by two main chips in the machine, these being the infamous Video ULA and the 6845 video controller.

Unfortunately it is not possible to test such a circuit in the same definitive way as with the sideways ROMs in last month's article. Instead a program is provided that attempts to use each of the facilities provided and it is then up to you to decide whether each feature is working correctly or is at fault.

To assist with this, each individual test is described, while certain more common faults are covered in detail to aid more conclusive testing. The tests carried out by the program can only be assessed subjectively and there may well be variations between different machines.

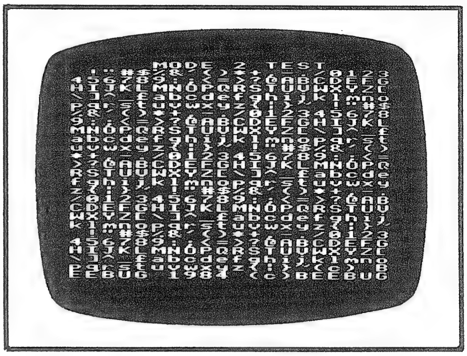
The program this month is written entirely in Basic and can be used on a 16k or 32k machine. The 16k test will take less time since fewer graphics modes are available.

<u>TESTING</u>

The program should be loaded and run. With a 16k machine the test will start at Mode 4, and at Mode Ø with a 32k system.

The Mode Ø test starts by filling the screen with characters and then scrolling first left and then right. The Video ULA is most prone to failure in the eighty column modes. This is normally evident by small speckles appearing randomly on the screen, or individual characters flickering.

Following the 'MODE Ø TEST', Mode 1 is tested. This follows the same format as the previous test but is in forty column mode. It is unlikely that this



or the twenty column tests will suffer from flickering. In each case the characters should be clearly defined and the scrolling smooth, moving first to the left and then back to the right.

These tests progress through Modes 2, 3, 4, 5 and 6, with the 16k tests starting at MODE 4. At the end of each test it is necessary to press the space bar in order to continue.

Following the Mode 6 test, a colour background test is included. In each case the background should be of the colour stated on the screen, and a steady picture maintained. In this test it is quite possible that poor colour quality could be due to poor tuning of a television, or a fault in the monitor or television being used. The main objective of the test is to ensure that the correct colours are produced and that the edges of the screen are clearly defined. If problems arise and a television is being used, try retuning the receiver to see if the picture improves.

The 'BLACK-FLASH' and 'WHITE-FLASH' tests are designed to show up faults in the video ULA. Look carefully at the vertical edges of the screen and ensure these are sharp. Make sure as well that the screen is either all black or all white and no lines are left of the other colour. In such cases the Video ULA will probably need replacing.

The final test is in Mode 7 and each colour is printed in columns, rather like a television test card. The order of the colours is written in the top line of the screen and should correspond to the bars. Each colour should be clearly defined and the screen steady. You can then terminate the program by pressing space bar.

Clearly, to use these tests fully does require the use of a colour display, and ideally an RGB monitor for the best results. Certain of the more common problems will be discernible on black and white or green and black screens, but the colour checks will obviously not be possible.

```
   1Ø REM VIDEO TEST PROGRAM
   2Ø REM Version B1.1
   3Ø REM Author Hugh D.Brown-Smith
   4Ø REM BEEBUG April 1984
   5Ø REM Program subject to copyright
   6Ø :
  1ØØ ON ERROR GOTO126Ø
  11Ø MODEØ:PRINTTAB(35,2)"MODE Ø TEST"
  12Ø FORX%=ØTO21
  13Ø FORY%=32TO126:VDUY%:NEXT
  14Ø NEXT
  15Ø PRINT'"(c) BEEBUG 1984 (c) BEEBUG
 1984 (c) BEEBUG 1984 (c) BEEBUG 1984 (
c) BEEBUG 1984"
  16Ø PROCmove(&3ØØØ)
  17Ø MODE1:PRINTTAB(15,2)"MODE 1 TEST"
  18Ø FORX%=ØTO1Ø
  19Ø FORY%=32 TO126:VDU Y%:NEXT Y%
  2ØØ NEXT X%
  21Ø PRINT"(c) BEEBUG  1984  (c)  BEEB
UG  1984 (c)"
  22Ø PROCmove(&3ØØØ)
  23Ø MODE2:PRINTTAB(5,2)"MODE 2 TEST"
  24Ø FORX%=ØTO4
  25Ø FORY%=32 TO126:VDU Y%:NEXT Y%
  26Ø NEXT X%
  27Ø PRINT"(c) BEEBUG 1984 (c)BEEBUG"
  28Ø PROCmove(&3ØØØ)
  29Ø MODE3:PRINTTAB(35,2)"MODE 3 TEST"
  3ØØ FORX%=ØTO16
  31Ø FORY%=32 TO126:VDU Y%:NEXT Y%
  32Ø NEXT X%
  33Ø PRINT"(c) BEEBUG 1984 (c) BEEBUG
1984 (c) BEEBUG 1984 (c) BEEBUG 1984"
  34Ø PROCmove(&4ØØØ)
  35Ø MODE4:PRINTTAB(15,2)"MODE 4 TEST"
  36Ø FORX%=ØTO1Ø
  37Ø FORY%=32 TO126:VDU Y%:NEXT Y%
  38Ø NEXT X%
  39Ø PRINT"(c) BEEBUG 1984 (c) BEEBUG
1984 (c) BEEBUG 1984"
  4ØØ PROCmove(&58ØØ)
  41Ø MODE5:PRINTTAB(5,2)"MODE 5 TEST"
  42Ø FORX%=ØTO4
  43Ø FORY%=32 TO126:VDU Y%:NEXT Y%
  44Ø NEXT X%
  45Ø PRINT"(c) BEEBUG 1984 (c) BEEBUG
1984"
  46Ø PROCmove(&58ØØ)
  47Ø MODE6:PRINTTAB(15,2)"MODE 6 TEST"
  48Ø FORX%=ØTO6
  49Ø FORY%=32 TO126:VDU Y%:NEXT Y%
  5ØØ NEXT X%
  51Ø PRINT"(c) BEEBUG 1984 (c) BEEBUG
1984 (c) BEEBUG 1984"
  52Ø PROCmove(&6ØØØ)
  53Ø MODE5:VDU19,Ø,1;Ø;
  54Ø PRINTTAB(4,1Ø)"RED BACKGROUND"
  55Ø PROCget:PROCblank:VDU19,Ø,2;Ø;
  56Ø PRINTTAB(2,1Ø)"GREEN BACKGROUND"
  57Ø PROCget:PROCblank:VDU19,Ø,3;Ø;
  58Ø PRINTTAB(2,1Ø)"YELLOW BACKGROUND"
  59Ø PROCget:PROCblank:VDU19,Ø,4;Ø;
  6ØØ PRINTTAB(3,1Ø)"BLUE BACKGROUND"
  61Ø PROCget:PROCblank:VDU19,Ø,5;Ø;
  62Ø PRINTTAB(1,1Ø)"MAGENTA BACKGROUND"
  63Ø PROCget:PROCblank:VDU19,Ø,6;Ø;
  64Ø PRINTTAB(3,1Ø)"CYAN BACKGROUND"
  65Ø PROCget:PROCblank:VDU19,Ø,7;Ø;
  66Ø COLOUR1:PRINTTAB(2,1Ø)"WHITE BACK
GROUND"
  67Ø PROCget:PROCblank:VDU19,Ø,8;Ø;
  68Ø PRINTTAB(4,5)"BLACK FLASH"
  69Ø PROCget:PROCblank
  7ØØ MODE5:COLOUR 135:CLS:VDU19,7,8;Ø;
  71Ø COLOUR1:PRINTTAB(4,5)"WHITE FLASH"
  72Ø PROCblank:PROCget
  73Ø MODE7
  74Ø PRINT" WHIT CYAN MAGE BLUE YELL G
REE RED  BLAC"
  75Ø FORX%=&7C28TO&8ØØØSTEP4Ø
  76Ø FORY%=ØTO35STEP5
  77Ø X%?Y%=(135-Y%/5)
  78Ø X%?(Y%+1)=157
  79Ø NEXT Y%,X%
  8ØØ FORX%=&7C4CTO&8ØØØSTEP4Ø
  81Ø ?X%=156
  82Ø NEXT X%
  83Ø PROCget
  84Ø MODE7:PRINT''"Test complete."
  85Ø END
  86Ø :
 1ØØØ DEFPROCscroll
 1Ø1Ø A%=(M%/8) MOD 256
 1Ø2Ø B%=(M%/8) DIV 256
 1Ø3Ø VDU23;12,M% DIV 2Ø48;Ø;Ø;Ø
```

»

```
 1040 VDU23;13,M% MOD 2048 DIV 8;0;0;0
 1050 T=TIME:REPEAT:UNTIL TIME>T+10
 1060 ENDPROC
 1070 :
 1080 DEFPROCmove(m%)
 1090 M%=m%
 1100 REPEAT M%=M%+8
 1110 PROCscroll
 1120 UNTIL M%=m%+&80
 1130 REPEAT M%=M%-8:PROCscroll:UNTIL M%
=m%
 1140 PROCget
 1150 ENDPROC
 1160 :
 1170 DEFPROCget
 1180 *FX15,1
 1190 A=GET
 1200 ENDPROC
 1210 :
 1220 DEFPROCblank
 1230 PRINTTAB(0,10);SPC20
 1240 ENDPROC
 1250 :
 1260 IF ERR=25 GOTO 350
 1270 ON ERROR OFF:MODE 7
 1280 IF ERR<>17 REPORT:PRINT" at ";ERL
 1290 END
```

NEWS NEWS NEWS

PROGRAMMING AIDS

Peter Bamford and Co. are distributing a really useful aid for anyone programming the BBC micro. This is called Pixel-Pad and consists of a set of large (A3) size screen layout charts giving full information on character positions and graphics co-ordinates for all graphics modes (0,1,2,4,5), together with 12 grids for designing user-defined characters. Also included is a similar sized card summarising some of the frequently needed information for the Beeb (such as modes, colours, VDU commands etc).

The Pixel-Pad costs £4.95 inc VAT (discounts and bumper packs available for education) from Peter Bamford and Co., 10 Barley Mow Passage, Chiswick, London W4 4PH.

LOW COST LOUDSPEAKER KIT

Capital City Electric have produced a loud speaker kit for the BBC micro. This comprises a loudspeaker in a smart black casing fitted with lead and jack plug plus a volume control, jack socket and wiring for fitting to your micro. No soldering is required and the whole operation takes but a few minutes [literally - Ed.] to install. The system was tried with a variety of music programs and gave reasonable sound quality. At £16.50 inclusive this seems good value for money. Contact Capital City Electric, 124 Arthur Rd, London SW19 8AA.

BUFFERS FOR THE BEEB

A>Line supply a range of printer buffers for Epson printers, and other general purpose buffers. They allow you to use your computer while printing is in progress. The buffers come in a range of sizes, from 2k to 64k. Prices range from £65 to £177.50 + VAT. Further details from:
A>Line Computer Systems, 1 Church Farm Lane, Willoughby Waterleys, Leicestershire, LE8 3UD.

TWO EPROM PROGRAMMERS

Two new EPROM programmers have been released recently. The first one is produced by Oasis and costs £199 (+VAT). It is capable of programming 24/28 pin EPROMS/EEPROMS and comes with the necessary programming software. OASIS can be found at: Oasis Electronics Ltd., University Village, Norwich, NR4 7TJ.

A much cheaper programmer is available from Softlife for 2764 and 27128 type EPROMS, which are the main EPROMs the BEEB can use. It connects onto the User Port, from which it takes its power. The programming software is suplied on EPROM, which must be loaded into memory via the programmer, and saved on disc or tape. It costs £59 (+VAT and post) from Softlife Ltd., 87 Silvertown Way, London, E16 4AH.

EDUCATIONAL SOFTWARE

Shiva Publishing Ltd. have decided to enter the ever expanding field of educational software, and have released the first of their software on numeracy and logic for primary schools. Further details can be obtained from Mr.N.John, 4 Church Lane, Nantwich, Cheshire, CW5 5RQ

Tested on Basic I & II and O.S. 1.2

MACHINE CODE GRAPHICS (part 3)

by Peter Clease

In the third instalment of our series on machine code graphics, we deal with multi-coloured graphics in Mode 2.

In the last two articles we introduced the principles of screen memory for Modes 5 and 4, and gave example programs for plotting both single and multi-sized multi-coloured characters in Mode 5. We now present details of the Mode 2 screen layout, and show you how to design and place multi-coloured characters on the screen in this mode.

MODE 2 GRAPHICS

Mode 2 is the most used mode when it comes to arcade games, probably because it offers the greatest number of colours at one time, and most arcade games are based around multi-coloured characters. The disadvantage of this mode is that it uses 2Øk of memory compared with the 1Øk of Mode 5 that we described last month. In Mode 2, sixteen logical colours are available, and each of these is represented by a group of four bits. Table 1 shows the relationship between the binary code and the associated logical colour.

Binary	Decimal	Logical Colour
ØØØØ	Ø	Black
ØØØ1	1	Red
ØØ1Ø	2	Green
ØØ11	3	Yellow
Ø1ØØ	4	Blue
Ø1Ø1	5	Magenta
Ø11Ø	6	Cyan
Ø111	7	White
1ØØØ	8	Black/White
1ØØ1	9	Red/Cyan
1Ø1Ø	1Ø	Green/Magenta
1Ø11	11	Yellow/Blue
11ØØ	12	Blue/Yellow
11Ø1	13	Magenta/Green
111Ø	14	Cyan/Red
1111	15	White/Black

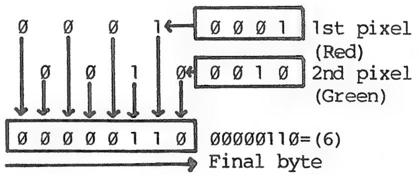
Table 1. Binary codes for colours in Mode 1.

This time four bits are needed to code the colour for each pixel (since 2 to the power 4 is 16, the number of colours to be specified) and, since each byte contains 8 bits, it follows that two pixels are represented by each byte. This leads to the byte map shown in figure 1 for the first character in a Mode 2 screen.

&3ØØØ	&3ØØ8	&3Ø1Ø	&3Ø18
&3ØØ1	&3ØØ9	&3Ø11	&3Ø19
&3ØØ2	&3ØØA	&3Ø12	&3Ø1A
&3ØØ3	&3ØØB	&3Ø13	&3Ø1B
&3ØØ4	&3ØØC	&3Ø14	&3Ø1C
&3ØØ5	&3ØØD	&3Ø15	&3Ø1D
&3ØØ6	&3ØØE	&3Ø16	&3Ø1E
&3ØØ7	&3ØØF	&3Ø17	&3Ø1F

Figure 1. Screen memory addresses of first character in Mode 2.

To illustrate the way that pixels are converted into bytes for Mode 2, let us produce a striped character block. The pixels on each line will be coloured red, green, yellow, blue, magenta, cyan, white and black respectively, working from left to right. Figure 2 shows how eight coloured pixels are converted into four bytes of data.



The other bytes code as follows:

Yellow and Blue......	26
Magenta and Cyan.....	53
White and Black......	42

Figure 2. Coding of pixels in Mode 2.

The following program (Program 7) will take the data calculated, and place it into the correct memory locations at the start of Mode 2 screen memory (&3000) to give us the desired result.

```
   10 REM PROGRAM 7
   20 REM AUTHOR PETER CLEASE
   30 REM BEEBUG APRIL 1984
   40 REM VERSION B1.0
   50 REM PROGRAM SUBJECT TO COPYRIGHT.
   60 :
  100 PROCassemble
  110 FORI%=0 TO 31
  120 READ I%?data
  130 NEXT
  140 DELAY=INKEY 200
  150 MODE 2:VDU23,1,0,0;0;0;0;
  160 CALL start
  170 PRINT'''
  180 END
  190 :
 1000 DEF PROCassemble
 1010 DIM data 31
 1020 DIM code 100
 1030 FOR PASS=0 TO 3 STEP 3
 1040 P%=code
 1050 [
 1060 OPT PASS
 1070 .start
 1080 LDY #0
 1090 LDA #0
 1100 STA &72
 1110 STA &70
 1120 LDA #&30
 1130 STA &71
 1140 .loop
 1150 LDA data,Y
 1160 STA (&70),Y
 1170 INY
 1180 CPY #32
 1190 BNE loop
 1200 RTS
 1210 ]
 1220 NEXT
 1230 ENDPROC
 1240 :
 2000 DATA 6,6,6,6,6,6,6,6
 2010 DATA 28,28,28,28,28,28,28,28
 2020 DATA 54,54,54,54,54,54,54,54
 2030 DATA 42,42,42,42,42,42,42,42
```

Note that the data is again listed DOWN each of the four columns of bytes in turn that comprise a character block in Mode 2. See earlier programs on screen access in this series for other examples.

TECHNICAL NOTES ON PROGRAM 7

Program 7 makes use of an addressing mode called indirect indexed addressing. For example, STA (&70),Y will store the contents of the accumulator in the memory location pointed to by &70 (low byte), and &71 (high byte) plus the current value in the Y register. In other words, &70 and &71 contain an address, and the contents of the Y register are added to this value to give the final memory address, at which the contents of the accumulator will be stored.

For example, if values are stored so that

```
   (&70)=&13         {Low byte}
   (&71)=&50         {High Byte}
    (Y) =3
   (Accumulator)=5
```

where (X) is a notation meaning 'the contents of X', then executing the instruction

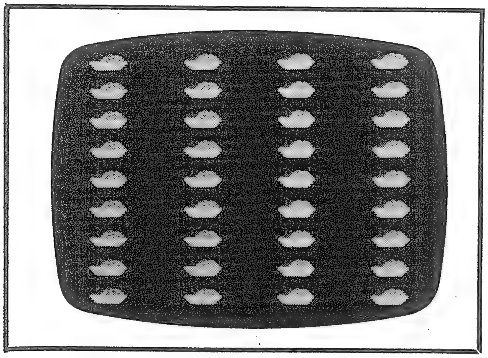
```
   STA (&70),Y
```

will result in 5 (from the accumulator) being stored in address &5016 (&50*&100+&13+3).

Two points to note: whenever the 6502 processor refers to a 2 byte address, as it does here, it ALWAYS expects it in the order low byte followed by high byte. Secondly, you can only use the indirect indexed addressing mode with the Y register, and with the indirect address residing in zero page.

Last month we listed a program (Program 6) to produce a shape that was two by two characters in Mode 5. This month we have listed a similar program (Program 8) that produces a shape of the same size but displays it in Mode 2. The process of designing each of the four characters is similar to that for the striped character block described earlier. This month we have chosen a bowl of ice-cream to make a change from the more usual space invader. The four characters are placed on the screen in the order top left, top right, bottom left, bottom right as in last month's example.

Once again, the program is written in the form of two procedures, PROCassemble, to assemble the machine code, and PROCdraw to place the shape on the screen at a given screen address. Although the data in our program produces a bowl of ice cream, you should be able to design your own shapes as well. The main program section also includes a loop (lines 150 to 200) which will display this colourful shape in a series of different positions on the screen.



```
  10 REM PROGRAM 8
  20 REM VERSION B1.4
  30 REM BEEBUG MARCH 1984
  40 REM AUTHOR DAVID FELL
  50 REM PROGRAM SUBJECT TO COPYRIGHT.
  60 :
 100 PROCassemble
 110 FOR I=0 TO 127
 120 READ data?I
 130 NEXT
 140 MODE 2
 150 FOR I=&3000 TO &7800 STEP 1920
 160 PROCdraw(I)
 170 PROCdraw(I+160)
 180 PROCdraw(I+320)
 190 PROCdraw(I+480)
 200 NEXT
 210 PRINTTAB(0,30)
 220 END
 230 :
 240 DATA 0,0,0,0,0,0,0,5
 250 DATA 12,0,4,4,4,1,3,3
 260 DATA 0,8,4,1,1,3,7,7
 270 DATA 0,0,0,7,15,15,15,15
 280 DATA 0,0,15,7,15,15,7,15
 290 DATA 0,0,4,15,15,7,15,15
 300 DATA 0,0,0,10,15,15,15,7
 310 DATA 0,0,0,0,0,0,10,10
 320 DATA 15,60,63,20,21,0,0,0
 330 DATA 15,60,63,60,63,60,63,20
 340 DATA 15,60,63,60,63,60,63,60
 350 DATA 11,60,63,60,63,60,63,60
 360 DATA 15,60,63,60,63,60,63,60
 370 DATA 7,60,63,60,63,60,63,60
 380 DATA 15,60,63,60,63,60,63,40
 390 DATA 7,60,63,40,42,0,0,0
 400 :
1000 DEF PROCassemble
1010 DIM CODE 200
1020 DIM data 127
1030 FOR PASS= 0 TO 3 STEP 3
1040 P%=CODE
1050 [
1060 OPT PASS
1070 LDY #0
1080 .loop1
1090 LDA data,Y
1100 STA (&80),Y
1110 INY
```

»»

TECHNICAL NOTES ON PROGRAM 8

In Program 8 the procedure PROCassemble assembles the machine code into an area of memory allocated from Basic via the DIM statement (line 1010). The procedure PROCdraw calls the assembled machine code to transfer the four characters to screen memory. Memory locations &80 and &81 point to the area in screen memory where the characters are to be displayed. The data produces, as was said earlier, a bowl of ice cream, but you could easily substitute the data for your own design.

Note the way in which the value of the loop counter (I at line 150) is transferred from the Basic program (as the parameter of PROCdraw in lines 160 to 190), where it is easy to manipulate and control loops, to the machine code (via line 2010), which actually places the character onto the screen. With careful planning, you could write a game in Basic, with most of the graphics handled by machine code.

An offset is needed between the plotting of the second and third characters, as these are on different lines on the screen, as explained in the 'Technical Notes' section of article number two in this series. Because of the difference in the amount of memory used to store the screen in Mode 2, the offset is &280, not &140 as it was for Mode 5. In the assembly section, lines 1070 to 1130 plot the first two characters, lines 1140 to 1200 add in the offset, and then lines 1210 to 1270 plot the second set of two characters onto the screen.

```
1120 CPY #64
1130 BNE loop1
1140 CLC
1150 LDA &80
1160 ADC #&40
1170 STA &80
1180 LDA &81
1190 ADC #2
1200 STA &81
1210 .loop2
1220 LDA data,Y
1230 STA (&80),Y
1240 INY
1250 CPY #128
1260 BNE loop2
1270 RTS
1280 ]
1290 NEXT
1300 ENDPROC
1310 :
2000 DEF PROCdraw(P%)
2010 !&80=P%
2020 CALL CODE
2030 ENDPROC
```

Next month, we will introduce some of the techniques necessary for simple animation and movement of characters on the screen.

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

PROBLEM WITH MASTERFILE USING AMCOM DFS

On an Acorn DFS, if a file is opened with on OPENIN or OPENUP command, and the file does not exist, the channel number is returned as zero, and no error is generated. With the AMCOM DFS, zero is returned, and an error is generated.

This anomaly causes problems with Masterfile, but has been corrected on later versions of the PACE DFS. On receipt of one of the older versions, PACE will upgrade the DFS free of charge to the A8639 version.

ROM CLASHES

Several ROMs on the market now have similar commands which may produce totally different actions. For example, *EDIT in the Watford DFS allows you to edit the disc sectors, in Disc Doctor it allows you to edit the function key definitions, and in BEEBUGSOFT's TOOLKIT it allows easy editing of a Basic program. If you have all three ROMs fitted in your machine at once, then the above command will be intercepted by the ROM in the highest numbered socket, which may not be the desired function.

In BEEBUG Vol.2 No.8, it was suggested how any ROM may be turned off. However a partial answer to the problem would be to place TOOLKIT in a ROM socket of lower value than either of the other two. This is because all of TOOLKIT's commands may be preceeded with 'B', e.g. *BEDIT, uniquely defining TOOLKIT's editing facility. This leaves a conflict of commands between only two ROMs!

DFS QUIRKS

With a 0.90 Acorn DFS it is possible to ask for a catalogue of a disc and obtain an incorrect result. The best way to illustrate this is to insert an unformatted disc and type *CAT <return>. While the disc is still spinning type *CAT <return> again. The DFS will return with 'Disc error' and then execute the second *CAT command. Since the disc is still spinning the DFS assumes that it already knows the catalogue, and will print out a copy of the catalogue presently held in memory, which will be a blank directory for an unformatted disc.

TRANSPARENT KEY DEFINITIONS - T.G.Ward

It is possible to load your definitions for the red function keys without corrupting a program which already exists in memory. This is acheived by *LOADing a set of definitions into the function key buffer and will not interfere with any other parts of memory. The definitions will have been set up at an earlier stage using the usual Basic program, and the buffer area from &B00 to &BFF is saved using *SAVE<name> B00 +100 where <name> is the filename for the function key definitions. [See BEEBUG Vol.1 No.4 for a more detailed explanation of function key use.]

FURTHER USES OF *BASIC

The OSBYTE call 187 can be used to enter any lanuage using the *BASIC command. The parameter placed in the X register is normally set to the correct position for the Basic ROM, but X can take any value from 0 to 15, and should be the desired language ROM. For example, if EXMON is placed in socket 12 with Basic in a higher priority socket, *FX187,12 will cause the command *BASIC to enter EXMON.
[Can anybody think of a use for this? - Ed.]

POSTBAG POSTBAG POSTBAG POSTBAG POSTBAG POSTBAG

TELETEXT MODE UPDATE

Dear Sir,

I quite like your magazine (I must, having subscribed for two years) and my only complaint is too many space invader type programs - even though I also find them quite addictive.

The main point of this letter is to correct your article on Teletext Mode, where in Vol.2 No.6 says you cannot type in graphics characters directly from the keyboard. Rubbish! This can be achieved using Control and a Function-key.

D.Taylor

Reply:

R.T.Evans and D.McMillan have also written similarly. I still believe that I was correct in the sense I intended. Using Control and a Function-key, you can then enter any Teletext Mode graphics control character as shown in the table below. Pressing any key for a lower case character will then display a graphics character on the screen. The problem is knowing which graphics character will result. If you want to produce a particular graphics character you will usually need to look this up in the User Guide, find its ASCII code, check which is the corresponding alpha character, and then press that key. This does not seem to me to be particularly simple or direct.

The use of Control and a Function-key is shown in the table below.

Control &	ASCII	Function
f0	144	nil
f1	145	red graphics
f2	146	green graphics
f3	147	yellow graphics
f4	148	blue graphics
f5	149	magenta graphics
f6	150	cyan graphics
f7	151	white graphics
f8	152	conceal display
f9	153	contiguous graphics

Another point is that it is impossible for us to print Teletext control or graphics characters in the magazine, and thus we do not use these as such, preferring instead functions such as CHR$ and STRING$. By incorporating such Teletext control characters directly in the program, you can, for example, write programs with coloured REM statements.

Finally, Desmond Walsham has also pointed out a small but vital omission in the procedure PROCline from the 4th part of the Teletext Mode series (BEEBUG Vol.2 No.8), where on page 23, line 1420 should read:

```
   1420 VDU31,x1%-1,y%,145+c%
```

MASTERFILE LIMITED BY WATFORD DFS

Dear Sir,

I thought you might be interested in my experience using the disc version of BEEBUG's Masterfile in conjunction with Watford Electronics DFS.

I purchased Masterfile to handle a very extensive Church Membership Roll. The equipment I used was Watford DFS version 1.1 and Cumana's double-sided 40/80 track twin disc drives.

The file was to contain 400 records with a total file length of 123,600 bytes. On trying to initialise this file, I found that errors occurred part way through record 212 and all of record 213. The message was "File not initialised beyond this point" and it then occurred to me that this point was just beyond 65,508 bytes. After some testing I began to suspect the DFS.

I subsequently visited Watford and attempted to get my 1.1 version of the DFS replaced with the 1.3. After some argument I had to pay a further £3.45 for the replacement.

On trying to initialise my file using the new chip, I succeeded first time. The 1.1 version seems to have a "brainstorm" when extending a file beyond 65536 bytes (64K).

I was most upset to have to pay yet more money for a workable version of the filing system but perhaps you could warn other members of this incompatibility and thus save them hours of typing.

M.J.Herbert

Reply:

We have no further comment to add to Mr Herbert's letter though we would be interested to hear any response from Watford Electronics.

POSTBAG POSTBAG POSTBAG POSTBAG POSTBAG POSTBAG

PROBLEMS OF UPGRADING O.S.1.1 TO 1.2

Dear Sir,

For some months now, my model B has been "hanging up" on me occasionally. Perhaps others might benefit from my eventual discovery that this was due to enlargement of the O.S. ROM socket, leading to intermittent loss of contact.

I find it particularly annoying that this problem was caused by by the O.S. 1.1 issue ROM provided on a "piggy-back" board with large pins. As I see it, therefore, Acorn replaced an O.S. which was below specification with one that has caused physical damage to my computer!

Iain Cameron

Reply:

The position is unfortunately much as Iain Cameron describes. We would suggest two possible solutions, one to bend inwards slightly the pins of the O.S. 1.2 ROM so that the pins are sprung against the sides of the socket holes when inserted. We have always found that this solved the problem completely. Alternatively you could install an additional socket with larger size pins into the original position and then install the new ROM in that.

THE MICRO GETS IT WRONG AGAIN!

Dear Sir,

I was rather surprised by the output of the following program:

```
10 FOR X=Ø TO 2
20 PRINT X
30 NEXT
40 PRINT X
```

I get Ø, 1, 2, 3 displayed when I would have expected Ø, 1, 2, 2. This (or a similar) routine incorporated in a program resulted in me spending an hour looking for a non-existent error. I feel that other BBC micro owners may well benefit from this knowledge.

Mark Fenton

Reply:

This shows how detailed is the knowledge that you need about even quite simple Basic instructions. In BBC Basic, a FOR-NEXT loop initially sets the loop control variable (in this case X) to the starting value (here Ø). The instructions in the loop are then executed (always at least once) and the control variable incremented before comparing its new value with the terminating value. If the control variable exceeds this value, then the loop is terminated otherwise the process is repeated. If you follow this through for the example given you will see why the results produced are as they are. Unfortunately, versions of Basic on other micros might handle this situation differently, so it is something always to be checked with a new Basic.

BUT SERIOUSLY

Dear Sir,

I have been a subscriber since BEEBUG was a baby. It is excellent value and the work you put in regarding the rectification of the bugs in the early BBC computers was admirable. The games programs in the magazine and on your tapes are all excellent value for money.

Games are all very well, but no matter how good they are some of us tire of them. I'm sure I'm not alone in this and would like something that would be useful in running a home. Perhaps a series of articles could be used to provide us with programs for home accounting and finance, or alternatively could you review some of the programs that are already available?

You may wonder why I don't write something myself but at 61 these new techniques do not come easy to the mind that was trained in steam radio.

A.G.Campbell

Reply:

Mr Campbell and others who share his point of view will be pleased to know that we agree with his suggestions. We have made a start with the program for annual budgets in this month's issue and we hope to publish more articles along these lines in the future together with reviews of useful applications software.

Tested on Basic I & II
and O.S. 1·2

ELEVASION(32K)

by D. J. Pilling

In the January/February issue of BEEBUG we published a superb fast action game called 'Block Blitz', which was well received by BEEBUG readers. This month, the author of Block Blitz, D.J.Pilling, brings you 'Elevasion', a fast game of skill which is even better, if that's possible.

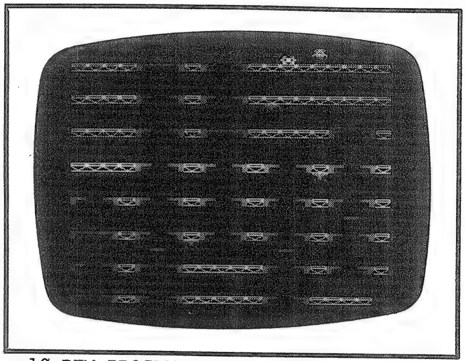
Elevasion is a one player game, in which you control a man running left and right along a network of corridors, or going up and down in various lifts. The object of the game is to kill as many 'droids' as possible by luring them under mines. When this happens, the droids are immoblised for a short time, allowing you to run underneath and kill them. If you fail to kill the droid, it will be released and come after you once again. If a second droid passes a droid that has already run into a mine, both droids will be released.

If the man passes underneath a mine, he is halted for a few seconds - enough time for the droids to catch him.

After five sheets of play, the mines will start to be replaced by 'blue cloners'. These have the terrifying effect of doubling a droid if it passes underneath. (i.e. there are now two droids instead of one).

The game requires the use of four keys, Z and X for left and right respectively, while * and ? control the direction of the lift (although the lift does also work automatically). You will also find that if the man disappears off one side of the screen, he will reappear at the other.

This is an outstanding game, particularly for one written in Basic. We have tried to keep the length of the program as reasonable as possible and hence the program listed does not display any of the instructions on the screen after the initial title page. The version of Elevasion on this month's magazine cassette is the full version which includes two additional screen pages showing the instructions described above.



```
  10 REM PROGRAM ELEVASION
  20 REM AUTHOR  D.J.Pilling
  30 REM VERSION B1.Ø
  40 REM BEEBUG  APRIL 1984
  50 REM PROGRAM SUBJECT TO COPYRIGHT
  60 :
 100 ON ERROR GOTO 240
 110 MODE7:VDU23;11,Ø;Ø;Ø;Ø
 120 PROCFP
 130 MODE5:PROCCHAR
 140 DIM A$(1),B$(1),C$(1)
 150 DIMX%(6),Y%(6),U%(6),L%(6),D%(6),
A%(6),B%(6),Z%(6),W%(6),I%(6),P%(6),U$(
6),D$(6),L$(1),R$(1),M$(1),A%33Ø
 160 PROCS
 170 REPEAT:PROCST
 180 REPEAT:PROCMA
 190 REPEAT:PROCL:PROCD:PROCM:UNTILOV%
 200 PROCG
 210 UNTIL ME%=Ø:PROCNXG
 220 UNTIL FALSE
 230 :
 240 ON ERROR OFF
 250 MODE 7:IF ERR=17 END
 260 REPORT:PRINT " at line ";ERL
 270 END
 280 :
1000 DEFPROCMA
1010 CLS:CF%=FALSE:VDU23;11,Ø;Ø;Ø;Ø
1020 VDU19,3,6,Ø,Ø,Ø
1030 PROCRM:PROCGM
1040 COLOUR2
1050 FORI%=3TO27STEP4:PRINTTAB(Ø,I%)ST
RING$(2Ø,CHR$24Ø);:NEXT:PRINTTAB(Ø,31)S
TRING$(19,CHR$24Ø);
```

»

```
 1060 FORI%=A%TOA%+287STEP21:$I%=STRING
$(20,CHR$7)+CHR$0:NEXT
 1070 A%?166=0:COLOUR1
 1080 FORI%=1TO6:FORJ%=U%(I%)TOL%(I%)
 1090 IF (J%+1)MOD4=0 COLOUR1:PRINTTAB(
X%(I%)-1,J%)CHR$252LB$CHR$253;:COLOUR1
ELSEPRINTTAB(X%(I%),J%)LB$;
 1100 K%=((J%+1)DIV4-1)*21+A%+X%(I%)
 1110 ?K%=I%:NEXT:I%(I%)=0
 1120 PRINTTAB(X%(I%),Y%(I%))L$;:NEXT
 1130 COLOUR3
 1140 FORI%=1TOND%
 1150 PROCSA
 1160 ?H%=8:PRINTTAB(J%,K%*4+2)CHR$246;
 1170 A%(I%)=J%:B%(I%)=K%*4+2
 1180 Z%(I%)=A%+K%*21:W%(I%)=0
 1190 NEXT
 1200 FORI%=1TONM%:PROCSM:NEXT
 1210 IFNC%=0ENDPROC
 1220 COLOUR3
 1230 FORI%=1TONC%
 1240 PROCSA
 1250 ?H%=10:PRINTTAB(J%,K%*4)CHR$242;
 1260 NEXT
 1270 ENDPROC
 1280 :
 1290 DEFPROCSA
 1300 J%=RND(18):K%=RND(7):H%=A%+J%+K%*
 21
 1310 IF?H%<>7OR?(H%-21)<>7OR?(H%+1)<>7
 OR?(H%-1)<>7OR?(H%-20)<>7OR?(H%-22)<>7
 GOTO1300
 1320 ENDPROC
 1330 :
 1340 DEFPROCSM:PROCSA:?H%=9:COLOUR1:PR
 INTTAB(J%,K%*4)CHR$242;:COLOUR3:ENDPROC
 1350 :
 1360 DEFPROCRM
 1370 X%=0:Y%=1:M%=1:Z%=A%:W%=FALSE
 1380 ENDPROC
 1390 :
 1400 DEFPROCGM
 1410 X%(1)=FNRX:U%(1)=3:L%(1)=U%(1)+4+
 RND(4)*4
 1420 X%(2)=FNRX:IFX%(2)=X%(1)GOTO1420
 1430 L%(2)=31:U%(2)=L%(1)-RND(2)*4
 1440 FORI%=3TO6
 1450 X%(I%)=FNRX:K%=0
 1460 FORJ%=1TOI%-1
 1470 IFX%(I%)<>X%(J%)GOTO1490
 1480 IFK%=0K%=J%ELSEK%=-1
 1490 NEXT
 1500 IFK%=0PROCNL:GOTO1550
 1510 IFK%=-1GOTO1450
 1520 U%(I%)=U%(K%)-3:L%(I%)=31-L%(K%)
 1530 IFU%(I%)<12ANDL%(I%)<12GOTO1450
 1540 IFU%(I%)<L%(I%)PROCNB ELSE PROCNT
 1550 NEXT
 1560 FORI%=1TO6:U$(I%)=MD$:D$(I%)=MD$:
 U$(I%)=LU$:D$(I%)=LD$:D%(I%)=RND(2)-1:Y
 %(I%)=(U%(I%)+L%(I%))/2:NEXT
 1570 ENDPROC
 1580 :
 1590 DEFPROCNL
 1600 U%(I%)=-1+RND(5)*4:L%(I%)=U%(I%)+
8+4*RND(3):IFL%(I%)>31L%(I%)=31
 1610 ENDPROC
 1620 :
 1630 DEFPROCNT
 1640 U%(I%)=-1+RND(2)*4:L%(I%)=U%(I%)+
4+RND(2)*4
 1650 IFL%(I%)>=U%(K%)GOTO1640
 1660 ENDPROC
 1670 :
 1680 DEFPROCNB
 1690 L%(I%)=35-RND(2)*4:U%(I%)=L%(I%)-
4-4*RND(2)
 1700 IFU%(I%)<=L%(K%)GOTO1690
 1710 ENDPROC
 1720 :
 1730 DEFFNRX=4*RND(5)-3
 1740 :
 1750 DEFPROCL
 1760 COLOUR1
 1770 FORI%=1TO6
 1780 IFD%(I%)GOTO1810
 1790 PRINTTAB(X%(I%),Y%(I%))U$(I%);:Y%(
I%)=Y%(I%)-2:IFY%(I%)=U%(I%)D%(I%)=TRUE
 1800 NEXT:ENDPROC
 1810 PRINTTAB(X%(I%),Y%(I%))D$(I%);:Y%
(I%)=Y%(I%)+2:IFY%(I%)=L%(I%)D%(I%)=FAL
SE
 1820 NEXT:ENDPROC
 1830 :
 1840 DEFPROCCHAR
 1850 VDU23,240,255,129,255,129,66,36,2
4,255
 1860 VDU23,241,255,255,255,0,0,0,0,0
 1870 VDU23,242,255,255,60,24,24,90,66,
66
 1880 VDU23,243,60,90,126,60,36,126,255
,189
 1890 VDU23,244,189,189,60,36,36,36,100
,6
 1900 VDU23,245,189,189,60,36,36,36,38,
96
 1910 VDU23,246,195,66,126,219,219,126,
66,195
 1920 VDU23,247,66,66,231,66,66,231,66,
66
 1930 VDU23,248,204,51,204,51,204,51,20
4,51
 1940 VDU23,252,255,255,255,224,224,224
,224,224
 1950 VDU23,253,255,255,255,7,7,7,7,7
 1960 VDU23,224,28,48,62,20,28,60,126,1
89
 1970 VDU23,225,189,60,24,24,56,104,200
,76

 1980 VDU23,226,189,60,24,24,28,23,18,24
 1990 VDU23,227,189,60,24,24,28,23,18,48
```

»

```
 2000 VDU23,228,189,60,24,24,56,104,200,
88
 2010 VDU23,229,28,6,62,20,28,60,126,189
 2020 VDU23,230,102,66,126,219,219,126,
66,102
 2030 ENVELOPE1,1,-15,-15,-15,230,230,2
30,30,5,0,-10,126,126
 2040 ENVELOPE2,3,0,1,0,0,255,0,127,0,0
,-127,126,126
 2050 ENVELOPE3,3,0,0,0,0,255,0,127,0,0
,-127,126,126
 2060 ENVELOPE4,1,-50,-50,-50,20,-20,20
,20,4,0,-5,120,110
 2070 ENVELOPE5,1,6,0,-6,200,100,200,10
0,2,0,-1,120,110
 2080 ENDPROC
 2090 :
 2100 DEFPROCS
 2110 TE$=CHR$10+CHR$8+CHR$8
 2120 ET$=CHR$8+CHR$10:EE$=CHR$8+CHR$11
 2130 EG$=CHR$11+CHR$8:R$=CHR$17+CHR$1
 2140 Y$=CHR$17+CHR$2:W$=CHR$17+CHR$3
 2150 B$=CHR$32+CHR$10+CHR$8+CHR$32
 2160 L$=CHR$241:LB$=CHR$32
 2170 LU$=LB$+CHR$11+EG$+L$
 2180 LD$=LB$+CHR$10+CHR$10+CHR$8+L$
 2190 MU$=LB$+EG$+LB$+EE$+L$+EE$+R$+CHR
$244+Y$+EE$+CHR$243+R$
 2200 MD$=LB$+Y$+EE$+CHR$11+LB$+ET$+LB$
+ET$+CHR$243+ET$+R$+CHR$244+ET$+R$+L$
 2210 TP$=Y$+CHR$8+CHR$229+LB$+TE$+R$:L
$(0)=TP$+CHR$227+LB$:L$(1)=TP$+CHR$228+
LB$
 2220 TP$=Y$+LB$+CHR$224+TE$+LB$+R$:R$(
0)=TP$+CHR$225:R$(1)=TP$+CHR$226
 2230 TP$=Y$+CHR$243+CHR$10+CHR$8+R$:M$
(0)=TP$+CHR$244:M$(1)=TP$+CHR$245
 2240 A$(0)=CHR$8+CHR$246+LB$
 2250 A$(1)=CHR$8+CHR$230+LB$
 2260 B$(0)=LB$+CHR$246
 2270 B$(1)=LB$+CHR$230
 2280 C$(0)=CHR$246:C$(1)=CHR$230
 2290 P$=A$(0):S$=B$(0):C$=C$(0)
 2300 DU$=LB$+EG$+LB$+EE$+L$+EE$+W$+CHR
$246+R$
 2310 DD$=LB$+W$+EE$+LB$+ET$+CHR$10+CHR
$246+ET$+R$+L$
 2320 CH$=R$+CHR$42+Y$+CHR$42+W$+CHR$42
 2330 CV$=Y$+CHR$42+ET$+R$+CHR$42+ET$+W
$+CHR$42+ET$
 2340 ?A%=0:A%=A%+1
 2350 HSC%=0:N$="JOE ZERO"
 2360 ENDPROC
 2370 :
 2380 DEFPROCM
 2385 FOR MV%=1TO500:NEXT
 2390 IFW%GOTO2480
 2400 IFM% M%=0ELSEM%=1
 2410 P%=Z%+X%:IF?P%>7PROCE2:ENDPROC
 2420 IFINKEY-67GOTO2440ELSEIFINKEY-98G
OTO2460
 2430 PRINTTAB(X%,Y%)M$(M%):ENDPROC
 2440 IFP%?1>6PRINTTAB(X%,Y%)R$(M%):X%=
X%+1ELSEL%=P%?1:PROCEN
 2450 ENDPROC
 2460 IFP%?-1>6PRINTTAB(X%,Y%)L$(M%):X%
=X%-1ELSEL%=P%?-1:PROCEN
 2470 ENDPROC
 2480 IFW%<0GOTO2550
 2490 IFINKEY-67 TX%=1:PROCEX:ENDPROC
 2500 IFINKEY-98 TX%=-1:PROCEX:ENDPROC
 2510 Y%=Y%(L%)
 2520 IFINKEY-73D%(L%)=0 ELSEIFINKEY-10
5D%(L%)=1 ELSEENDPROC
 2530 IFY%(L%)=U%(L%)D%(L%)=TRUE ELSEIF
Y%(L%)=L%(L%)D%(L%)=FALSE
 2540 ENDPROC
 2550 W%=W%+1:IFW%=0:PROCSM
 2560 ENDPROC
 2570 :
 2580 DEFPROCEN
 2590 IFL%=0GOTO2660
 2600 IFY%(L%)<>Y%+2ORI%(L%)PRINTTAB(X%
,Y%)M$(M%):ENDPROC
 2610 SOUND1,2,100,2:PRINTTAB(X%,Y%)B$
 2620 U$(L%)=MU$:D$(L%)=MD$:X%=X%(L%)
 2630 PRINTTAB(X%,Y%)M$(M%)
 2640 W%=1:I%(L%)=9
 2650 ENDPROC
 2660 IFX%=0PRINTTAB(0,Y%)B$:X%=19+(Y%=
29):PRINTTAB(X%,Y%)M$(M%):ENDPROC
 2670 IFX%>17 PRINTTAB(X%,Y%)B$:X%=0:PR
INTTAB(X%,Y%)M$(M%):ENDPROC
 2680 PRINTTAB(X%,Y%)M$(M%)
 2690 ENDPROC
 2700 :
 2710 DEFPROCEX
 2720 IF(Y%(L%)+1)MOD4=0ELSEENDPROC
 2730 SOUND1,2,200,2
 2740 U$(L%)=LU$:D$(L%)=LD$:W%=FALSE
 2750 I%(L%)=0
 2760 Y%=Y%(L%)-2:Z%=((Y%(L%)+1)DIV4-1)
*21+A%
 2770 PRINTTAB(X%,Y%)B$:X%=X%+TX%
 2780 PRINTTAB(X%,Y%)M$(M%):ENDPROC
 2790 :
 2800 DEFPROCD
 2810 IFE% E%=0ELSEE%=1
 2820 P$=A$(E%):S$=B$(E%):C$=C$(E%)
 2830 COLOUR3
 2840 FORD%=1TO5
 2850 IFD%>ND%:PROCDT(0):NEXT:ENDPROC
 2860 IFW%(D%)GOTO2940
 2870 U%=A%(D%):V%=Z%(D%)+U%:S%=B%(D%)
 2880 IFS%=Y%+1P%(D%)=U%<X%:IFU%=X%NEXT
:ENDPROC
 2890 IFP%(D%)GOTO2920
 2900 IFV%?-1>6ELSET%=V%?-1:PROCDE:NEXT
:ENDPROC
 2910 ?V%=7:PRINTTAB(U%,S%)P$:A%(D%)=U%
-1:IFV%?-1<9V%?-1=8:NEXT:ENDPROC ELSEPR
OCE1M:NEXT:ENDPROC
```

»

```
 2920 IFV%?1>6ELSET%=V%?1:PROCDE:NEXT:E
NDPROC
 2930 ?V%=7:PRINTTAB(U%,S%)S$:A%(D%)=U%
+1:IFV%?1<9V%?1=8:NEXT:ENDPROC ELSEPROC
E1P:NEXT:ENDPROC
 2940 IFW%(D%)<0GOTO3020
 2950 R%=Y%-Y%(W%(D%))+2
 2960 IFR%=0PROCDX:NEXT:ENDPROC
 2970 IFR%<0GOTO3000
 2980 IFNOTD%(W%(D%))PROCDX
 2990 NEXT:ENDPROC
 3000 IFD%(W%(D%))PROCDX
 3010 NEXT:ENDPROC
 3020 W%(D%)=W%(D%)+1
 3030 IFW%(D%)=0PROCRE:SOUND3,5,200,10
 3040 NEXT:ENDPROC
 3050 :
 3060 DEFPROCDE
 3070 IFS%=Y%(T%)-1ELSEIFT%=0P%(D%)=NOT
P%(D%):ENDPROC ELSEPRINTTAB(U%,S%)C$:EN
DPROC
 3080 IFS%+1=U%(T%)ORS%+1=L%(T%)GOTO3130
 3090 IFY%<=S%GOTO3120
 3100 IFNOTD%(T%)ENDPROC
 3110 GOTO3130
 3120 IFD%(T%)ENDPROC
 3130 IFI%(T%)ENDPROC
 3140 ?V%=7
 3150 SOUND1,2,100,2:PRINTTAB(U%,S%)LB$
 3160 U$(T%)=DU$:D$(T%)=DD$:A%(D%)=X%(T
%)
 3170 PRINTTAB(A%(D%),S%)C$
 3180 W%(D%)=T%:I%(T%)=D%
 3190 ENDPROC
 3200 :
 3210 DEFPROCDX
 3220 IF(Y%(W%(D%))+1)MOD4=0ELSEENDPROC
 3230 SOUND1,2,200,2
 3240 T%=W%(D%):W%(D%)=0:I%(T%)=0
 3250 U$(T%)=LU$:D$(T%)=LD$:B%(D%)=Y%(T
%)-1
 3260 PRINTTAB(A%(D%),B%(D%))LB$
 3270 IFY%=B%(D%)-1  P%(D%)=A%(D%)<X%
 3280 IFP%(D%)  A%(D%)=A%(D%)+1ELSEA%(D%
)=A%(D%)-1
 3290 PRINTTAB(A%(D%),B%(D%))C$(E%)
 3300 Z%(D%)=((Y%(T%)+1)DIV4-1)*21+A%:?
(Z%(D%)+A%(D%))=8
 3310 ENDPROC
 3320 :
 3330 DEFPROCE1M
 3340 U%=U%-1:V%=V%-1:GOTO3380
 3350 :
 3360 DEFPROCE1P
 3370 U%=U%+1:V%=V%+1
 3380 IF?V%=10CF%=TRUE
 3390 IF?V%>10  EX%=D%:D%=?V%-10:SOUND1,
1,200,10:PROCRE:D%=EX%:A%(D%)=A%(D%)+1:
?V%=8:ENDPROC
 3400 ?V%=10+D%:SOUND1,1,200,10:SOUND1,
5,200,10:SOUND2,2,200,30
```

```
 3410 FOREX%=1TO36:COLOUREX%MOD4
 3420 PRINTTAB(U%,B%(D%));:VDU246,8,11,
247,8,11,242
 3430 NEXT
 3440 IFCF%PROCCL:ENDPROC
 3450 COLOUR2:PRINTTAB(U%,B%(D%)-2)CHR$
246:COLOUR3:W%(D%)=-50
 3460 ENDPROC
 3470 :
 3480 DEFPROCEA:IFD%=ND%GOTO3500ELSEA%(
D%)=A%(ND%):B%(D%)=B%(ND%):W%(D%)=W%(ND
%):Z%(D%)=Z%(ND%)
 3490 IFW%(ND%)<0 ?(Z%(ND%)+A%(ND%))=10
+D%
 3500 ND%=ND%-1:D%=9:DD%=DD%+1:IFND%=0O
V%=TRUE
 3510 ENDPROC
 3520 :
 3530 DEFPROCRE:PRINTTAB(A%(D%),B%(D%)-
2)LB$;:VDU8,10,10,246:W%(D%)=0:PROCSM:E
NDPROC
 3540 :
 3550 DEFPROCER:D%=?P%-10:?P%=7:COLOUR1
:PRINTTAB(A%(D%),B%(D%)-2)CHR$246:COLOU
R3:PROCEA:SOUND0,3,6,20:ENDPROC
 3560 :
 3570 DEFPROCE2
 3580 IF?P%>10PROCER:ENDPROC
 3590 IF?P%>8W%=-20:?P%=7:PRINTTAB(X%,Y
%-1)LB$:SOUND1,4,200,20:ENDPROC
 3600 SOUND1,4,200,20:PROCDT(25)
 3610 PRINTTAB(X%,Y%)CHR$248:PRINTTAB(X
%,Y%+1)CHR$248:PROCDT(35):PRINTTAB(X%,Y
%)B$
 3620 ME%=ME%-1:DD%=DD%+1:OV%=TRUE
 3630 ENDPROC
 3640 :
 3650 DEFPROCDT(I%):T=TIME:REPEATUNTILT
IME>T+I%:ENDPROC
 3660 :
 3670 DEFPROCG
 3680 CLS:SC%=SC%+250+200*DD%
 3690 IFSC%DIV2000<>BC% ME%=ME%+1
 3700 IFME%=0ENDPROC
 3710 VDU19,3,2,0,0,0
 3720 PRINTTAB(1,1)STRING$(6,CH$)TAB(0,
1)STRING$(10,CV$)TAB(19,1)STRING$(10,CV
$)TAB(0,30)Y$+CHR$42+STRING$(6,CH$)+Y$+
CHR$42;
 3730 PRINTTAB(5,2)Y$+"ELEVASION"TAB(4,
4)"SCORE BOARD"TAB(4,6)"FOR SHEET ";G%;
TAB(1,7)STRING$(6,CH$);
 3740 PRINTTAB(2,9)"YOUR SCORE ";SC%
 3750 PRINTTAB(2,11)"HIGH SCORE ";HSC%
 3760 PRINTTAB(2,14)R$"DROIDS DESTROYED"
 3770 X%=DD%:IFX%>8 X%=8
 3780 PRINTTAB(2,16)STRING$(X%,CHR$246+
CHR$9)
 3790 PRINTTAB(2,20)Y$"MEN LEFT"
 3800 X%=ME%:IFX%>8 X%=8
 3810 PRINTTAB(2,22)STRING$(X%,M$(0)+CH
R$11+CHR$9);
 3820 X%=1
 3830 IFSC%DIV2000<>BC%ANDX%=1PRINTTAB(
2,25)Y$"BONUS MAN" ELSEPRINTTAB(2,25)SP
C(10)
 3840 PRINTTAB(1,28)R$"RETURN TO CONTIN
UE"
 3850 IFINKEY-74GOTO3870
 3860 IFTIME MOD15=0 X%=X%EOR1:GOTO3830
 ELSEGOTO3850
 3870 VDU19,3,6,0,0,0:OV%=FALSE
 3880 G%=G%+1:DD%=0:BC%=SC%DIV2000
 3890 IFG%<=5GOTO3930
 3900 ND%=G%-5:IFND%>5ND%=5
 3910 NC%=G%-5:IFNC%>8NC%=8
 3920 NM%=2+ND%:ENDPROC
 3930 ND%=G%:NM%=2+ND%
 3940 ENDPROC
 3950 :
 3960 DEFPROCST
 3970 G%=1:ND%=1:NM%=3:NC%=0:DD%=0:ME%=
3:OV%=FALSE:SC%=0:BC%=0
 3980 ENDPROC
 3990 :
 4000 DEFPROCCL
 4010 CF%=FALSE:SOUND1,4,200,20
 4020 IFND%=5ENDPROC
 4030 ND%=ND%+1:PROCSM
 4040 A%(ND%)=A%(D%)+1:B%(ND%)=B%(D%):Z
%(ND%)=Z%(D%):W%(ND%)=0
 4050 ENDPROC
 4060 :
 4070 DEFPROCNXG
 4080 CLS:VDU19,3,2,0,0,0
 4090 PRINTTAB(1,1)STRING$(6,CH$)TAB(0,
1)STRING$(10,CV$)TAB(19,1)STRING$(10,CV
$)TAB(0,30)Y$+CHR$42+STRING$(6,CH$)+Y$+
CHR$42;
 4100 PRINTTAB(5,2)Y$+"ELEVASION"TAB(4,
4)"GAME   OVER";TAB(1,5)STRING$(6,CH$);
 4110 PRINTTAB(3,7)"THE HISCORE WAS"TAB
(8,9);HSC% TAB(3,11)"BY "N$
 4120 PRINTTAB(3,14)Y$"YOUR SCORE WAS "
TAB(8,16);SC%
 4130 IFSC%<=HSC% GOTO4240
 4140 PRINTTAB(3,18)"THE NEW HISCORE"
 4150 PRINTTAB(3,21)R$"ENTER YOUR NAME"
 4160 *FX21,0
 4170 PRINTTAB(3,23)"-->"Y$;
 4180 N$="":REPEAT IK=GET
 4190 IFIK=13GOTO4220
 4200 IFIK=127ANDLENN$<>0PRINTCHR$IK;:N
$=LEFT$(N$,LENN$-1):GOTO4220 ELSEIFIK=1
27 GOTO4220
 4210 N$=N$+CHR$IK:PRINTCHR$IK;
 4220 UNTIL IK=13OR LENN$=12
 4230 HSC%=SC%
 4240 PRINTTAB(4,26)R$"PRESS RETURN"TAB
(2,28)"FOR ANOTHER GAME";:PROCDT(100):R
EPEAT UNTILINKEY-74
 4250 ENDPROC
 4260 :
 4270 DEFPROCFP
 4280 CLS:FORA=2TO3:PRINTTAB(12,A)CHR$1
41;CHR$130"ELEVASION":NEXT
 4290 PRINTTAB(16,12);CHR$134"by"
 4300 PRINTTAB(12,14)CHR$134"D.J.Pillin
g"
 4310 PRINTTAB(8,20)"Press any key to s
tart."
 4320 G=GET:CLS
 4330 ENDPROC
```

NOTE: If you find the game too fast for you, it can be slowed down by inserting a delay as line 2385. For example, try

```
   2385 TIME=0:REPEAT UNTIL TIME>5
```

Values other than 5 will give different delays.

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

RECYLING PRINTER RIBBONS - R.Skemp

When you think that your printer ribbon has come to the end of its useful life, do not throw it away immediately, because Richard Skemp suggests prising the lid off the cartridge, lightly spraying the ribbon with WD-40 (available from all good motoring shops!) and leaving overnight. He has tried it on an MX80 ribbon, and the result appears to be the same as replacing with a new ribbon.

EPSON SCREEN DUMP UPDATE - A.Wilmshurst

The Epson 8-tone screen dump on the Vol.2 No.4 magazine cassette prints tones corresponding to the logical colours on the screen and not the actual colours, causing problems if the pallette is changed. The program may be altered to print actual colours as follows:

```
580 LDA #&B:LDX #(osblock+4) MOD 256:LDY #(osblock+4) DIV 256:JSR osword:PLA:TAX:LDA
 osblock+5:AND #7:STA point,X
```

The resulting program should be *SAVEd from &2A00 to &2BD4.

Tested on Basic I & II and O.S. 1.2

DARTBOARD(32K)

by J. Crombie

Have you ever wondered how top dart players such as Eric Bristow and Jocky Wilson reach the heights of perfection when they play? Well, BEEBUG provides one answer, would you believe, with a program that will help you play better than ever before, and maybe score that fabulous one hundred and eighty!

Dartboard is a game for two players (though you can easily take both parts if you wish) using either the keyboard or joysticks to control your aim. If you choose to use joysticks then you may still use the keyboard for fine adjustments.

This is the traditional game of darts where each player takes turns to throw three darts to reduce his score down from the starting value of 501, finishing on a double or a 'Bulls-eye'.

You may use either a joystick or the keyboard (selected at the start of the match) to aim a dart before pressing a number key (1 to 9) to throw the dart and to select the relative strength of the throw, 1 being the weakest and 9 being the strongest.

The keys for controlling direction are 'Z' and 'X' for left and right, and '*' and '?' for up and down.

As you throw your darts the computer will automatically count up the total of your three darts and display your required score at the right-hand side of the screen.

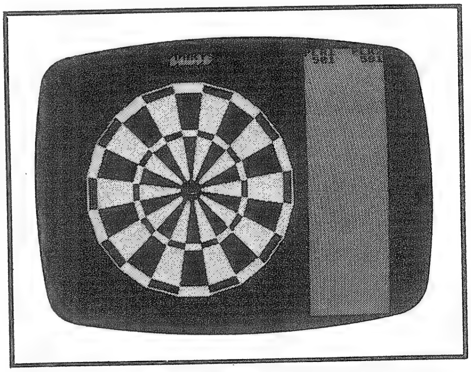
When a game has finished you can choose either to play another game or, by pressing Return, to see the result of the match displayed on the screen.

PROGRAM NOTES

The program is quite well structured using a number of clearly named procedures to do most of the work. These procedures are listed below with a brief description of the function of each.

1000 PROCboard	Draws dartboard on screen.
1180 PROCsector	Draws one sector of the dartboard.
1270 PROCmovesight	Moves target point for dart using joystick or keyboard.
1490 PROCdart	Throws dart.
1590 PROCtotal	Calculates dart score.
1730 PROCremovedarts	Removes darts from board.
1810 PROCendgame	Displays match result.
1940 PROCkeyorjoy	Displays front page and selects joystick or keyboard control.
2070 PROCinit	Initialise sound envelopes, user-defined characters and variables.

The main program loop is from line 180 to 310 which is repeated until one player is the winner.



```
 10 REM PROGRAM DARTS
 20 REM AUTHOR  J.Crombie
 30 REM VERSION B1.0
 40 REM BEEBUG  APRIL 1984
 50 REM PROGRAM SUBJECT TO COPYRIGHT
 60 :
100 ON ERROR GOTO 2120
110 MODE7:PROCkeyorjoy
```

»

```
  12Ø MODE1:VDU5:DIM SCORE%(2Ø),SUM(2),
DX%(3),DY%(3),game%(2)
  13Ø PROCinit
  14Ø PROCboard
  15Ø REPEAT
  16Ø COLOURØ:PRINT"PLR1  PLR2 5Ø1    5Ø
1"
  17Ø COLOUR3:VDU28,3Ø,31,39,1
  18Ø REPEAT
  19Ø TIME=Ø:Player%=Player%+1
  2ØØ IF Player%=3 Player%=1
  21Ø sum%=Ø:dart%=1
  22Ø REPEAT
  23Ø winner%=FALSE
  24Ø PROCmovesight(dart%)
  25Ø IF SUM(Player%)=sum% AND double%=
1 THEN winner%=TRUE:GOTO31Ø
  26Ø IF SUM(Player%)-sum%<2 THEN VDU4:
PRINTTAB(6*(Player%-1)+1);"---  ";:VDU5
:PROCremovedarts(dart%):winner%=TRUE:GO
TO31Ø
  27Ø dart%=dart%+1
  28Ø UNTIL dart%>3
  29Ø PROCremovedarts(3)
  3ØØ SUM(Player%)=SUM(Player%)-sum%:VD
U4:PRINTTAB(6*(Player%-1)+1);SUM(Player
%);SPC(2);:VDU5
  31Ø UNTIL winner%
  32Ø VDU4:SOUND1,1,5Ø,1:PRINT''''''" W
INNER     Player";Player%''''''''''':gam
e%(Player%)=game%(Player%)+1
  33Ø PRINT" PRESS KEY "
  34Ø A=GET:IF A=13 PROCendgame:END
  35Ø IF dart%=4 THEN PROCremovedarts(3
) ELSE PROCremovedarts(dart%)
  36Ø VDU28,3Ø,31,39,Ø,12:Player%=2:SUM
(1)=5Ø1:SUM(2)=5Ø1
  37Ø UNTIL FALSE
  38Ø END
  39Ø :
 1ØØØ DEF PROCboard
 1Ø1Ø FORI%=1 TO 2Ø
 1Ø2Ø PROCsector(I%,3,5ØØ,5ØØ,4Ø6)
 1Ø3Ø PROCsector(I%,3*(I% MOD 2),5ØØ,5Ø
Ø,4ØØ)
 1Ø4Ø PROCsector(I%,3*(ABS((I% MOD 2)-1
)),5ØØ,5ØØ,365)
 1Ø5Ø PROCsector(I%,3*(I% MOD 2),5ØØ,5Ø
Ø,225)
 1Ø6Ø PROCsector(I%,3*(ABS((I% MOD 2)-1
)),5ØØ,5ØØ,2ØØ)
 1Ø7Ø NEXT
 1Ø8Ø FOR I%=1 TO 2Ø:PROCsector(I%,Ø,5Ø
Ø,5ØØ,4Ø):PROCsector(I%,1,5ØØ,5ØØ,15):N
EXT
 1Ø9Ø FOR I%=1 TO 2Ø
 11ØØ MOVE45Ø*SIN(RAD(18*I%-16))+471,44
Ø*COS(RAD(18*I%-16))+5ØØ:READ A$:PRINTA$
 111Ø GCOLØ,2:MOVE43Ø+I%,1ØØØ+I%:PRINT"
DARTS":GCOLØ,1
```

```
 112Ø NEXT
 113Ø GCOLØ,Ø:MOVE43Ø+I%,1ØØØ+I%:PRINT"
DARTS"
 114Ø RESTORE:FOR I%=1 TO 2Ø:READ SCORE
%(I%):NEXT
 115Ø VDU4,28,3Ø,31,39,Ø,19,2,4,Ø,Ø,Ø:C
OLOUR13Ø:CLS
 116Ø ENDPROC
 117Ø :
 118Ø DEFPROCsector(N%,C%,X%,Y%,S%)
 119Ø GCOLØ,C%
 12ØØ MOVEX%,Y%
 121Ø MOVE S%*SIN(RAD(18*(N%-1)-9))+X%,
S%*COS(RAD(18*(N%-1)-9))+Y%
 122Ø PLOT85,S%*SIN(RAD(18*N%-9))+X%,S%
*COS(RAD(18*N%-9))+Y%
 123Ø ENDPROC
 124Ø :
 125Ø DATA 2Ø,1,18,4,13,6,1Ø,15,2,17,3,
19,7,16,8,11,14,9,12,5
 126Ø :
 127Ø DEFPROCmovesight(dartno%)
 128Ø VDU5:F%=Ø
 129Ø C%=A%:D%=B%
 13ØØ IF kb% GOTO 135Ø
 131Ø IF ADVAL(2*(Player%-1)+1)>5ØØØØ A
%=A%-4
 132Ø IF ADVAL(2*(Player%-1)+2)<8ØØØ B%
=B%-4
 133Ø IF ADVAL(2*(Player%-1)+1)<8ØØØ A%
=A%+4
 134Ø IF ADVAL(2*(Player%-1)+2)>5ØØØØ B
%=B%+4
 135Ø IF INKEY-98 A%=A%-4
 136Ø IF INKEY-67 A%=A%+4
 137Ø IF INKEY-73 B%=B%+4
 138Ø IF INKEY-1Ø5 B%=B%-4
 139Ø IF A%<Ø A%=Ø ELSE IF A%>9ØØ A%=9Ø
Ø
 14ØØ IF B%<Ø B%=Ø ELSE IF B%>9ØØ B%=9Ø
Ø
 141Ø IF TIME>3ØØ GOTO 144Ø
 142Ø *FX15,1
 143Ø GOTO145Ø
 144Ø ?6Ø2=32:KEY%=INKEY(Ø):IF KEY%>48
AND KEY%<58 THEN GCOL3,2:MOVEC%,D%:VDU2
24:PROCdart(A%,B%,dartno%):ENDPROC
 145Ø GCOL3,2:IF F%=1 MOVEC%,D%:VDU224
 146Ø F%=1:MOVEA%,B%:VDU224
 147Ø GOTO129Ø
 148Ø :
 149Ø DEFPROCdart(X%,Y%,dartno)
 15ØØ GCOL3,1
 151Ø X%=X%+16+RND(32*(KEY%-&3Ø))-8*(KE
Y%-&3Ø):Y%=Y%-12+RND(32*(KEY%-&3Ø))-8*(
KEY%-&3Ø)-12*(&3A-KEY%)
 152Ø MOVEX%-2Ø,Y%-2Ø:DRAWX%+2Ø,Y%+2Ø
 153Ø MOVEX%-2Ø,Y%+2Ø:DRAWX%+2Ø,Y%-2Ø
 154Ø DX%(dartno)=X%:DY%(dartno)=Y%
 155Ø SOUNDØ,-15,5,1
```

»

```
 156Ø PROCtotal(X%,Y%)
 157Ø ENDPROC
 158Ø :
 159Ø DEFPROCtotal(X%,Y%)
 16ØØ IFY%=5ØØ Y%=499+RND(2)
 161Ø ANGLE%=DEGATN((X%-5ØØ)/(Y%-5ØØ)):
radius%=SQR((X%-5ØØ)^2+(Y%-5ØØ)^2)
 162Ø IF Y%-5ØØ<Ø THEN ANGLE%=ANGLE%+18
Ø ELSE IF ANGLE%<-9 THEN ANGLE%=ANGLE%+
36Ø
 163Ø SEC%=(ANGLE%+9)/18
 164Ø DARTSCORE%=SCORE%(SEC%+1):double%
=Ø
 165Ø IF radius%<4Ø6 AND radius%>365 TH
EN DARTSCORE%=DARTSCORE%*2:double%=1
 166Ø IF radius%>2ØØ AND radius%<225 TH
EN DARTSCORE%=DARTSCORE%*3
 167Ø IF radius%>14 AND radius%<4Ø THEN
 DARTSCORE%=25
 168Ø IF radius%<15 THEN DARTSCORE%=5Ø:
double%=1
 169Ø IF radius%>4Ø6 THEN DARTSCORE%=Ø
 17ØØ sum%=sum%+DARTSCORE%
 171Ø ENDPROC
 172Ø :
 173Ø DEF PROCremovedarts(dartno%)
 174Ø GCOL3,1
 175Ø FORI%=1 TO dartno%
 176Ø MOVEDX%(I%)-2Ø,DY%(I%)-2Ø:DRAWDX%
(I%)+2Ø,DY%(I%)+2Ø
 177Ø MOVEDX%(I%)-2Ø,DY%(I%)+2Ø:DRAWDX%
(I%)+2Ø,DY%(I%)-2Ø
 178Ø NEXT
 179Ø ENDPROC
 18ØØ :
 181Ø DEFPROCendgame
 182Ø VDU22,7
 183Ø VDU133,157,13,1Ø
 184Ø VDU132,157,131,141:PRINTTAB(12);"
DARTS"
 185Ø VDU132,157,131,141:PRINTTAB(12);"
DARTS"
 186Ø VDU133,157,13,1Ø
 187Ø SOUND1,1,2Ø,1:SOUND2,1,32,1:SOUND
3,1,44,1
 188Ø PRINTTAB(5,12);"Player 1 has won
";game%(1);" games"
 189Ø PRINTTAB(5,14);"Player 2 has won
";game%(2);" games"
 19ØØ IF game%(1)=game%(2) THEN PRINTTA
B(1,2Ø);"I therefore declare the contes
t a draw"'':ENDPROC
 191Ø PRINTTAB(1,2Ø);"I therefore decla
re the winner ";:IF game%(1)>game%(2) T
HEN PRINT"Player 1" ELSE PRINT"Player 2
"
 192Ø ENDPROC
 193Ø :
 194Ø DEFPROCkeyorjoy
 195Ø VDU23;11,Ø;Ø;Ø;Ø
 196Ø FORA=2TO3:PRINTTAB(11,A)CHR$141;C
HR$131"Dartboard":NEXT
 197Ø PRINTTAB(11,6)CHR$134"by J.Crombi
e"
 198Ø PRINTTAB(9,12)CHR$13Ø"Are you pla
ying"''TAB(1Ø)CHR$13Ø"from"CHR$133"K"CH
R$13Ø"eyboard"
 199Ø PRINTTAB(12)CHR$13Ø"or"CHR$133"J"
CHR$13Ø"oystick"
 2ØØØ REPEAT:VDU7
 2Ø1Ø PRINTTAB(15,17)CHR$131"?"CHR$132
 2Ø2Ø ?6Ø2=32:G$=GET$:PRINTTAB(19,17)G$
 2Ø3Ø UNTIL G$="K" OR G$="J"
 2Ø4Ø IF G$="K" kb%=TRUE ELSE kb%=FALSE
 2Ø5Ø ENDPROC
 2Ø6Ø :
 2Ø7Ø DEFPROCinit
 2Ø8Ø ENVELOPE 1,149,2Ø,-1Ø,Ø,5,1,5,126
,Ø,Ø,-7,126,1ØØ
 2Ø9Ø VDU23,224,8,8,8,255,8,8,8,8:A%=5Ø
Ø:B%=5ØØ:SUM(1)=5Ø1:SUM(2)=5Ø1:@%=Ø:Pla
yer%=2:game%(1)=Ø:game%(2)=Ø
 21ØØ ENDPROC
 211Ø :
 212Ø ON ERROR OFF
 213Ø MODE 7
 214Ø IF ERR=17 END
 215Ø REPORT:PRINT" at line ";ERL
 216Ø END
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

DISC UPGRADE FOR MODEL A MICROS - P.Ireland

If you intend to upgrade a model A BBC micro to include a disc interface, without using an official upgrade, then do not forget to add IC77 (74LSØØ) for the disc upgrade, as it will not function without it. The list in the Advanced User Guide fails to mention this in its list of parts for the disc upgrade, because it is supplied in the model A to B conversion. (IC76 is also needed to read paged ROMs - see BEEBUG Vol.1 No.1Ø).

IF YOU WRITE TO US

BACK ISSUES (Members only)
All back issues are kept in print (from April 1982). Send 90p per issue PLUS an A5 SAE to the subscriptions address. This offer is for members only, so it is ESSENTIAL to quote your membership number with your order. Please note that the BEEBUG Reference Card and BEEBUG supplements are not supplied with back issues.

SUBSCRIPTIONS
Send all applications for membership, subscription renewals, and subscription queries to the subscriptions address.

MEMBERSHIP COSTS:
U.K.
£5.40 for 6 months (5 issues)
£9.90 for 1 year (10 issues)
Eire and Europe
Membership £16 for 1 year.
Middle East £19
Americas and Africa £21
Elsewhere £23
Payments in Sterling preferred.

Subscriptions & Software Address	Subscriptions and Software Help Line
BEEBUG	St.Albans
PO BOX 109	(0727) 60263
High Wycombe	Manned Mon-Fri
Bucks	1pm-4pm

PROGRAMS AND ARTICLES

All programs and articles used are paid for at around £25 per page, but please give us warning of anything substantial that you intend to write. In the case of material longer than a page, we would prefer this to be submitted on cassette or disc in machine readable form using "Wordwise", "Minitext Editor" or other means. If you use cassette, please include a backup copy at 300 baud.

HINTS

There are prizes of £5 and £10 for the best hints each month.

Please send all editorial material to the editorial address below. If you require a reply it is essential to quote your membership number and enclose an SAE.

Editorial Address
BEEBUG
PO Box 50
St Albans
Herts

BEEBUG MAGAZINE is produced by BEEBUG Publications Ltd.
Editor: Mike Williams.
Technical Editor: Philip Le Grand. Production Editor: Phyllida Vanstone.
Technical Assistants: Alan Webster and David Fell.
Managing Editor: Lee Calcraft.
Thanks are due to Sheridan Williams, Adrian Calcraft, John Yale, Robert Barnes, Michael Beasley, Hugh Brown-Smith and Tim Powys-Lybbe for assistance with this issue.

BEEBUG NEW ROM OFFER

1.2 OPERATING SYSTEM

A special arrangement has been agreed between Acorn and BEEBUG whereby BEEBUG members may obtain the 1.2 operating system in ROM at the price of £5.85 including VAT and post and packing.
The ROM will be supplied with fitting instructions to enable members to install it in their machine.
If the computer does not subsequently operate correctly, members may take their machine to an Acorn dealer for the upgrade to be tested, which will be done at a charge of £6.00 plus VAT. This charge will be waived if the ROM is found to have been defective. If the computer has been damaged during the installation process, the dealer will make a repair charge.

FREE

NEW ROMS FOR OLD
EXCHANGE YOUR 1.0 FOR THE 1.2

We can now exchange your old 1.0 operating system for the new 1.2, free of charge. To take advantage of this offer, please send your 1.0 (supplied on eprom with a carrier board), in good condition to the address below.

£5 FOR YOUR OLD 1.0

If you have the 1.0 operating system and have already bought a 1.2, we will exchange the 1.0 (supplied on eprom with a carrier board) for a £5 voucher. This voucher may be used against any purchase from BEEBUGSOFT.

ADDRESS FOR 1.2 OS:-
ROM Offer, BEEBUG, PO Box 109, High Wycombe, Bucks, HP11 2TD.

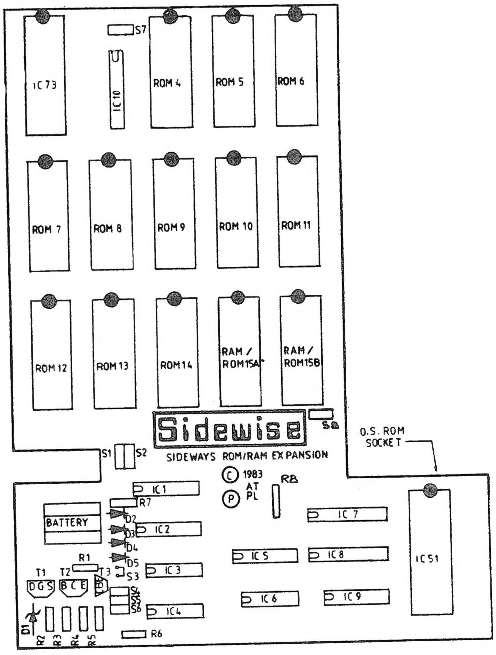
BEEBUGSOFT

ATPL'S SIDEWAYS ROM EXPANSION BOARD

SPECIAL PRICE TO MEMBERS £39.00 inc.

Save £5.70 on normal price of £44.70

* Simply plugs into the BBC Micro
* No soldering necessary
* Increases the sideways ROM capacity to 16
* Fully buffered - allows all sockets to be used
* Complete with full and detailed instruction booklet.
* Accepts 16K RAM in special sockets
* Battery back up facility for RAM (parts available directly from ATPL at extra cost)
* As used at BEEBUG
* Reviewed in BEEBUG vol.2 number 6



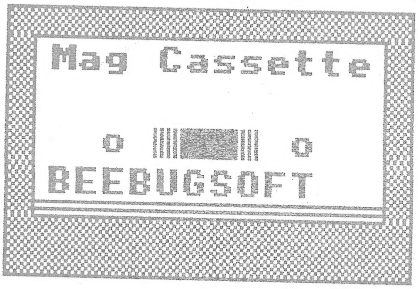
HOW TO ORDER

Please send your order with a cheque / postal order made payable to BEEBUG, and enclose your membership number. We are unable to supply the board to overseas members.

The address for SIDEWAYS is:
BEEBUGSOFT, PO Box 109, High Wycombe, Bucks. HP11 2TD.

MAGAZINE CASSETTE OFFER

To save wear and tear on fingers and brain, we offer, each month, a cassette of the programs featured in the latest edition of BEEBUG. The first program on each tape is a menu program, detailing the tape's contents, and allowing the selection of individual programs. The tapes are produced to a high technical standard by the process used for the BEEBUGSOFT range of titles. Ordering information, and details of currently available cassettes are given below.



All previous magazine cassettes (from Vol.1 No.10) are available.

This month's cassette (Vol.2 No.10) includes: Elevasion game, Machine Code Graphics example programs, Tape Indexing programs, Dartboard game, program to produce 3D Histograms, Home Accounting - Annual Budgets, Testing Out Your Micro (Part 2) - The Video ULA, Moving Text Display, April Fool Routine, plus Four Tone Mode 7 screen dumps for Epson FX80 and MX80 printers.

All magazine cassettes cost £3.00 each. For ordering information see BEEBUGSOFT advertisement at the back of this month's magazine supplement.

MAGAZINE CASSETTE SUBSCRIPTION

We are able to offer members subscription to our magazine cassettes. Subscriptions will be for a period of one year and are for ten consecutive issues of the cassette. If required, subsriptions may be backdated as far as Vol.1 No.10, which was the first issue available on cassette. This offer is available to members only, so when applying for subscription please write to the address below, quoting your membership number and the issue from which you would like your subscription to start.

CASSETTE SUBSCRIPTION ADDRESS:

Please send a sterling cheque with order, together with your membership number and the date from which the subscription is to run, to:
PO Box 109, High Wycombe, Bucks,

CASSETTE SUBSCRIPTION PRICE:
UK £33 inc VAT and p&p
OVERSEAS (inc Eire) £39 inc p&p
(no VAT payable).

BEEBUG BINDER OFFER

BEEBUG MAGAZINE BINDER OFFER

A hard-backed binder for BEEBUG magazine is available. These binders are dark blue in colour with 'BEEBUG' in gold lettering on the spine. They allow you to store the whole of one volume of the magazine as a single reference book. Individual issues may be easily added or removed, thus providing ideal storage for the current volume as well.

BINDER PRICE
U.K. £3.90 inc p&p, and VAT.
Europe £4.90 inc p&p
(no VAT payable).
Elsewhere £5.90 inc p&p
(no VAT payable).

Make cheques payable to BEEBUG.
Send to Binder Offer, BEEBUG, PO Box 109, High Wycombe, Bucks.
Please allow 28 days for delivery on U.K. orders.

Printed in England by Staples Printers St Albans Limited at The Priory Press.
ISSN 0263-7561